

DIGITAL COMPILATION OF

GEOLOGIC MAPPING

Prepared by:

L. Sue Beard, Debra Block, Melissa Aldrich, Haydee Hampton, and Anita Kaye
Western National Cooperative Geologic Mapping Team
Flagstaff, Arizona

December 1998

DRAFT

2255 N. Gemini Drive, Flagstaff, AZ 86001
520-556-7196, sbeard@flagmail.wr.usgs.gov
520-556-7138, dblock@flagmail.wr.usgs.gov

TABLE OF CONTENTS

I. INTRODUCTION	4
A. Stable-base layers for drafted geologic map and related information.....	4
B. Database layers for a geologic map	4
II. PROCEDURE FOR CREATING A DIGITAL GEOLOGIC MAP.....	6
A. Getting data into digital form.....	6
1. Digitizing.....	6
2. Gridline	6
3. ArcScan.....	7
B. Georeference	7
1. Register	8
2. Project	9
3. Transform.....	9
C. Preparation of <name_poly> coverage for editing.....	10
1. Bulk editing (for scanned maps to be used with gridline)	10
2. Creating INFO files	11
a. Symbol sets.....	11
b. Lookup tables	11
c. Relate files	14
d. Adding items to feature attribute tables.....	14
D. Editing and attributing <name_poly> layer	15
1. Introduction	15
2. Startup AMLs.....	16
3. Pulldown menus.....	20
4. Editing process	20
E. Structure line layer <name_fold>.....	21
1. Scan	21
2. Digitized cover.....	22
F. Structure point layer <name_dip>.....	22
1. Adding items.....	22
2. Adding attitudes.....	23
G. Annotation layer <name_anno>.....	24
1. Creating annotation.....	24
2. Manipulating annotation.....	24
III. TABLE SUMMARIZING DIGITAL STRUCTURE OF GEOLOGY COVERAGES	26

ATTACHMENTS

1. Markerset and lookup table – geo.mrk
2. Lineset and lookup table – geo.lin
3. Example of lookup table (lut) for polygon shades
4. Example of README file
5. newlut.aml
6. additempoly.aml
7. editarc.aml, editpoly.aml
8. arcedit.menu, polyedit.menu
9. tiptoe.aml
10. additemfold.aml
11. additemdip.aml
12. attitude.aml, attitude.menu, plunge.aml, dip.aml
13. additemanno.aml
14. announitedit.aml, announit.menu

I. INTRODUCTION

This document-in-progress presents current procedures used by the Flagstaff GIS Team to produce digital geologic maps. Our intent is to outline a basic routine that will guide the user through the necessary steps in creating a digital map from the analog information contained on the source map. The audience is assumed to have some familiarity with geologic maps, but we have prepared this document for a relatively new user of ARC/INFO. We do recommend that the user refer to ARC/INFO Online Help for detailed information on commands and techniques. Also, we are constantly revising and streamlining our techniques, and welcome all suggestions on this manual that will enhance our GIS lab. Please direct your comments to Sue Beard (tel: 520-556-7196; email: sbeard@usgs.gov) or Debra Block (tel: 520-556-7138; email: dblock@usgs.gov)

A. Stable-base layers for drafted geologic map and related information

There are many options available for preparing geologic information for digital format. We either digitize maps or scan and vectorize linework using Grid or ArcScan. The following items are optimal for both original mapping and for simplified 1:100,000 linework:

- Greenline of quadrangle – inked with linework only, using continuous thin lines (approx 3x 0 rapidograph). For scanning or digitizing.
- Punch-registered mylar overlay – inked with all other info to be digitized (fault ornamentation, unit labels, attitudes, sample locations).
- Punch-registered mylar overlay (optional) – any other specialized information to be added to the database.
- Geology generalized to 1:100,000 scale. We can send 1:24,000 mylar plots of the 1:100,000 DLG data for your particular quadrangle if you so choose. The plots can include hypsography, public lands, roads, and tic marks for registration to your original greenline (quadrangle). You can generalize the 1:24,000-scale geology by directly overlaying these plots on your original quadrangle. Pencil works well on these mylar plots (which are produced by the HP inkjet plotter). We can then digitize or scan the generalized geology. (The DLG data is printed in blue and therefore are dropped out in the scanning process.)
- Colored ozalid copy of map.
- Explanation – list of units, symbols, description of units if available. Ascii text file or Word file format is best. Please email the text file to sbeard@usgs.gov.
- Tabulated data – can be entered into database.

B. Database layers for a geologic map

The digital database for a geologic map is typically organized into layers, or coverages, according to theme and feature type. Digital layers separate different kinds of information or features that are readily combined for graphic or analytical purposes. Each coverage consists of topologically linked (topology is a mathematical method of defining spatial relationships using arc-node data structure) features and associated descriptive data.

<u>Geologic layer</u>	<u>Coverage</u>	<u>Description</u>
Contacts, faults, units	<name_poly>	All lines labeled by type and polygons labeled by unit name.
Structure (points)	<name_dip>	Bedding attitudes, lineations, etc.

Structure (lines)	<name_fold>	Fold axes, dikes, structural contours, cross-section lines.
Annotation	<name_anno>	Unit labels and leaders, fault names, other map-face text.
Other (points)		Wells, geochemistry, etc.
Other (lines)		Seismic profiles, etc.

For 1:100,000 scale mapping, we use USGS DLG vector data; for scales other than 1:100,000, we create raster layers from scans of topographic map separates. Data can be obtained by ftp from edcftp.cr.usgs.gov: pub/data/DLG/100K/. We process data using amls written by Marilyn Flynn and Tim Leiber, WRD. Other options include DEM and DRG bases.

<u>Base layer</u>	<u>Coverage</u>	<u>Description</u>
DLG	<name_hypso>	Topographic relief.
DLG	<name_hydro>	Trace of water bodies.
DLG	<name_rr>.	Railroads
DLG	<name_rd>	Roads
DLG	<name_misc>	Miscellaneous transportation
DLG	<name_bound>	Landuse boundaries.
DLG	<name_publand>	Township/range sections.
Raster	<name_contour>	Contour separate.
Raster	<name_drainage>	Drainages separate.
Raster	<name_misc>	Boundary, culture, etc. separate.

II. PROCEDURE FOR CREATING A DIGITAL GEOLOGIC MAP

NOTE: Add registration tics to greenline prior to digitizing or scanning. Use stick-on crosshairs. Use a minimum of four tics, preferably at or near margins. Small scale maps and those crossing UTM zones require more tics.

Create README text file and update throughout process (Ex.: ATTACHMENT 4). Include original source of data, parameter files used in conversion from raster to vector, mastertic coverage data, projection info, transform data, etc. Also keep informal logbook or readme file of daily processes.

Always make backup copy of coverage before altering:

Arc: copy <name_cover> <name_cover+date>

A. Getting data into digital form

Three different methods of transforming geologic maps into digital format are use (described below), depending on original sources of mapping:

1. **Digitizing**, using either an Altek or Summagraphics digitizer (in Flagstaff).
2. Scan and ‘bulk’ vectorize using ARC/INFO **gridline** command.
3. Scan and interactively vectorize using **ArcScan**.

1. Digitizing

No preliminary preparation needed, other than identification and numbering of tics (registration marks on maps). Read Lesson 4 in “*Understanding GIS: the ARCINFO Method*” for a good introduction to digitizing.

2. Gridline

a. Linework scanned in Flagstaff on Optronics 5040 raster scanner. Generally scanned at 50 microns (508dpi).

b. Scanner outputs raster file in RLE format <name.rle>.

c. Convert RLE format to RLC format <name.rlc>:

The scanner used in Flagstaff cannot output an image in an ARC/INFO readable format, so the file must be converted for use. Marilyn Flynn wrote a C program that can perform the conversion, however, it runs only on a Data General or a Sun (rigel.wr.usgs.gov). Copy program from /home/dblock/bin/rle2rlc to /home/(name)/bin.

%: rle2rlc (prompts for input and output)

d. Convert RLC format file to ARC/INFO Grid, <name_grid>:

Arc: imagegrid <name.rlc> <name_grid>

Draw to screen to check orientation. If necessary, can rotate grid.

AE: mapextent image <name_grid>

AE: image <name_grid>

AE: draw

If grid needs rotating:

Arc: gridrotate <name_grid> <name_gridrot> <angle>

- e. Vectorize file (FOR GRIDLINE ONLY, not ArcScan):

Arc: gridline <name_grid> (or <name_gridrot>) <name_vector>

If it bails, try thinning in grid. The parameters in the example below may need to be modified.

Grid: <name_th_grid> = thin<name_grid> (or <name_gridrot>), then

Arc: gridline <name_th_grid> <name_vector> # nothin ## # 10

Draw to screen, check quality of vectorization and look for elements that can be removed with batch editing (see Bulk Editing, page 10).

3. ArcScan

NOTE: We highly recommend that the user read the entire contents of the ArcScan section in ARC/INFO online help before proceeding. This guide was written based on ARC/INFO version 7.1

- a. Obtain geologic map (stable-base best), add tic marks and scan.
- b. Choose appropriate scanning parameters (i.e., dots per inch):
Map needs to be scanned at appropriate resolution for line work. ArcDoc recommends that raster lines are best traced in ArcScan when they are 4 to 6 pixels wide with a minimum of 3 pixels. First, determine the line thickness of the lines on your map. Then either:
 - i. Determine the resolution (dpi=dots per inch=pixels per inch) necessary to fit from 3-6 pixels per line (the greater the number of pixels per line the more time it takes to open a grid file, but the higher the precision of the grid and less likely there will be gaps in lines.) Scan at the calculated dpi.
 - ii. If your scanner has only a few resolution settings (the only appropriate resolutions the Flagstaff drum scanner has are 508dpi or 1016dpi), calculate the number of pixels given the line thickness at the scanner resolutions available.
- c. Scan map to create .rle file and convert .rle to .rlc file:
Our scanner creates run-length encoded (.rle) files. ARC/INFO will import run-length compressed files, but not RLE files, so in Flagstaff we run a conversion program written in C to convert them; called ‘rle2rlc’.
- d. Import .rlc file into ARC/INFO and convert file into a grid:
 - i. Open ArcTools from the arcedit prompt:
 - Arc: arctools
 - ii. Select ‘Edit Tools’ from the ‘ArcTools’ pulldown menu
 - iii. Select ‘Tools’ on the ‘Edit Tools’ pulldown menu and choose ‘Command Tools’
 - iv. Select ‘Conversion’ from the ‘Command Tools’ pulldown menu, choose ‘To Grid’, and then choose ‘Image to Grid’
 - v. Enter the name of your RLC file in the ‘Input image’ field and name the ‘Output grid’ <name_ascan>
 - vi. Select ‘Square’ as the ‘Blocking Method’
 - vii. Click OK to perform the conversion

NOTE: Before tracing lines, you can either transform the GRID into real world coordinates or trace first and then transform. The instructions below are for transforming first.

B. Georeference

For digitizing of standard 1:24,000, 1:62500, and 1:100,000 quadrangles, the *Alacarte* module (<http://wrgis.wr.usgs.gov/docs/software/software.html>) can be used to create the tic cover as well as

project and transform it. You can then copy the tic coverage, which contain the tics and boundary in projected real-world coordinates, to <name_poly> for digitizing. However, Alacarte won't list the transformation errors (RMS).

1. Register

Tics represent geographic control points used to register the spatial coordinates in a digital map.

- For a **digitized** coverage either create a tic cover using Alacarte or:
Define digital tics at crosshairs on the latitude-longitude graticule.
 - Arc: digitizer altek /dev/tty00:9600:8bit:none
[or digitizer sgm3/dev/tty01:9600:7bit:none]
 - Arc: arcedit
 - AE: coordinate digitizer
 - AE: create <name_dig> (returns screen prompt to digitize tics)
Digitize corner tics, clockwise from lower right. Be sure to place crosshair over center of tic mark. Can specify tic id while digitizing. (Use ‘coordinate digitizer <name>_dig’ to register map, and use ‘coordinate digitizer’ once map is registered during each edit session).
 - EXAMPLE:
On digitizer puck: 1 A 1 for first tic id, 2 A 1, second id, etc, O A to end. Prompts for boundary, draw on digitizer from slightly beyond lower left (xmin, ymin) to upper right (xmax, ymax).
 - AE: save
- For a **scanned** map, after using gridline (name_vector) or ArcScan (name_ascan):
 - AE: mape <name_vector> (or <name_ascan>)
 - AE: ec <name_vector> (or <name_ascan>)
 - AE: de arc tic id
 - AE: draw
 - AE: ef tic
 - AE: add (will enter menu – 1 on keyboard or left button on mouse will add tic, 9 on keyboard or CTL-right mouse button will end)
Add four corner tics at crosshairs on scanned image. Add additional internal tics if necessary. Use the move command to reregister the digital tic if it is not correctly placed.
 - AE save
- If **not** using Alacarte:
The tic coordinates are currently held in digitizer measurements (inches) or pixel space and must be converted to the same real-world coordinate system and projection as the source map. Before the tics can be related to projected coordinates they must be geographically referenced. Identify the degrees of latitude and longitude corresponding to the tic locations and convert to decimal degrees. An AML, deg2dec.aml, may be used to translate coordinates.
 - Create an empty coverage <name_tic> containing only the tic and boundary coordinates from the digitized or scanned coverage.
 - Arc: create <name_tic> <name_vector> (or <name_dig> or <name_ascan>)
 - Update tics to latitude-longitude coordinates in INFO.

```

Arc: info
*(Hint: INFO commands need to be typed in CAPS)
INFO: user name? ARC
INFO: SEL NAME_TIC.TIC
INFO: UPDATE XTIC,YTIC BY IDTIC PROMPT
INFO: IDTIC? Enter tic #
INFO: XTIC> Enter decdeg # for longitude
*(Hint: longitude numbers are recorded as negative values)
INFO: YTIC> Enter decdeg # for latitude
*Continue for all tics, to end hit enter at prompt.
INFO: Q STOP (to quit INFO)

```

2. Project

Project the tics from latitude-longitude coordinates to projected units. This procedure will create the working polygon coverage.

EXAMPLE:

```

Arc: project cover <name_tic> <name_poly>
Project: input
Project: projection geographic
Project: units dd
Project: parameters
Project: output
Project: projection utm
Project: units meters
Project: zone 11 (check in ArcDoc (help) if zone not known)
Project: parameters
Project: end

```

A text file containing the projection parameters may be used instead of performing the projection interactively.

3. Transform

Convert coverage created in digitizer or scanner units into real-world units. Be sure to save transform RMS error; use your text editor and copy this data into the README file.

- For a **digitized** coverage or a coverage vectorized with **gridline**:

```
Arc: transform <name_dig> (or name_vector) <name_poly>
```

RMS is calculated in pixel space on scanned covers; to recalculate for inches divide the input error by the dpi of scan:

EXAMPLE:

```
rms error (input, output) = (2.591, 8.108) [pixels, meters]
dpi of scan = 508, therefore 2.591/508 = (0.00510, 8.108) [inches, meters]
```

- For a coverage vectorized using **ArcScan**:

ArcScan is invoked in ArcTools, so the following directions will specify ArcTools menus.

- To transform the grid, first open the <name_tic> cover in ArcTools, ArcEdit under menu item “open coverages:”

- Set edit feature to arc

- Make sure the feature “tic” is set to on in "Display" menu under "Draw Env-General"
- Invoke “Trace Env”
- Open <name_ascan>
- Invoke “Edit”
- Invoke “Register”
- Zoom in or out to see tic marks
- Hit cross-hair button under "fit the entire grid within a box"
- Zoom in on tic closely using “Create a window” option, so can return to tic coverage easily
- Use “move the grid” option
- Invoke “Warp grid”
- Invoke “Set...” menu
 - Set snap link endpoints to tics.
 - Set reference coverage to tic coverage
- Invoke “Add under link operations” in “warp grid” menu
- Draw a link from tics on grid to tics in tic coverage (use mouse button 1 on grid and button 2 to snap link to tic cov tic.)
- Toggle “9” to stop adding and save linkfile by hitting Save
- Set “order of polynomial” to one to transform grid
- Invoke “RMS report” **IMPORTANT: Save report to README file.**
- If RMS error is acceptably low, invoke “warp”.
- Use the ARC "create" command to create a blank coverage (using the projected tic file) to trace arcs into

EXAMPLE:

Arc: *create <name_poly> <name_tic>*

2) For conversion to vector file:

- Set tracing parameters:
 - Set tracing parameters (see ArcDoc). Then save these parameters: under Save/Load, load *.vtp tracing properties file (you can incorporate .vtp files into menus).
 - Set Edit Env: make sure the Arc and Node snap tolerances are small enough to prevent lines and nodes from snapping together erroneously.
- Trace arcs:

Hit Trace and use mouse buttons (see ArcDoc) to trace arcs. As you trace, adjust tracing properties as needed. You can attribute the arcs while you are tracing by using table manager and table editor (see ArcDoc).

C. Preparation of <name_poly> coverage for editing

1. Bulk editing (for scanned maps to be used with gridline)

Two options for bulk editing:

- Edit scan (raster image, <name>_grid) prior to projection:

Arc: *scan_clean_50.aml*

Editing done in linesample/pixel space. Use amls created by Trent Hare in the /arcwork/atool directory. Please read usage and warnings before running.

- Edit after the scan is vectorized and projected (<name>_poly):
 - AE: mape <name_poly>
 - AE: ec <name_poly>
 - AE: de arc tic
 - AE: draw
 - AE: ef arc (or poly)

Editing done in meters, therefore scale specific. Sample small area and determine appropriate parameters.
Remove extraneous polygons by selecting for area, delete remnant tic vectors from scan, set grain for spline.

2. Creating INFO files

Our general procedure is to:

- *Move the lookup tables and relate files to the current workspace, using the newlut.aml.*
Arc: &r newlut.aml
- *Add items to the .pat and .aat for <name_poly>, using the additempoly.aml.*
Arc: &r additempoly.aml
- *Create the custom polygon lookup table for the geologic map.*

These steps are described in detail below.

a. Symbol sets

The geologic features and cartographic elements on a map are drawn using symbols. We generally use symbols chosen from three symbolsets that have been modified from symbols created at the USGS in Menlo Park:

<u>Symbolset</u>	<u>Description</u>	
geo.lin	lineset for symbolizing arcs	ATTACHMENT 1
geo.mrk	markerset for symbolizing arcmarkers and points	ATTACHMENT 2
alc1.shd	shadeset for coloring polygons	

b. Lookup tables

A lookup table is an INFO data file used to categorize item values and associate each with a symbol number. Each lookup table contains two items. One item is defined and named the same as an item in a feature attribute table and contains records for each unique attribute value. The other item is named ‘symbol’ and contains specific symbol numbers that correspond to each unique record. A lookup table must be present in the INFO directory of the current workspace in order to access. Because multiple records in the feature attribute table access a single symbol number in the lookup table, it is easy to universally change how an attributed feature is depicted.

Since we typically use a generic lineset and markerset for all of our geologic coverages, we likewise use standard lookup tables (geolin.lut, geomrk.lut, respectively) to symbolize arcs and markers. However, geolin.lut can be customized if line types on a map are not represented or if the name is not appropriate. A lookup table (name_poly.lut) assigning a color to each map unit (polygon) is generally created for each coverage. Before creating the lookup table it is helpful to make a list of the data to be entered, i.e., all unit names and their associated color number from the shadeset.

<u>Lookup table</u>	<u>Symbolsset</u>	
geolin.lut	geo.lin	ATTACHMENT 1
geomrk.lut	geo.mrk	ATTACHMENT 2
Name_poly.lut	alc1.shd	ATTACHMENT 3

- To create a lookup table in INFO for a *lineset*:

NOTE: This file is already created on hydra and transferred to the current workspace using newlut.aml. The following is for information only.

INFO:

```
ENTER COMMAND> DEFINE GEOLIN.LUT
    ITEM NAME> ltype
    ITEM WIDTH> 35
    ITEM OUTPUT WIDTH> 35
    ITEM TYPE> c
    ITEM NAME> symbol
    ITEM WIDTH> 3
    ITEM OUTPUT WIDTH> 3
    ITEM TYPE> i
```

This creates the .lut template. To add data to the info file:

```
ENTER COMMAND> ADD
    LTYPE> contact_certain
    SYMBOL> 16
```

NOTE: Data entry errors can be corrected with the UPDATE command or by using PURGE to delete selected records.

- To create a lookup table in INFO for a *markeriset*:

NOTE: This file is already created on hydra and transferred to the current workspace using newlut.aml. The following is for information only.

INFO:

```
ENTER COMMAND> DEFINE GEOMRK.LUT
    ITEM NAME> pttype
    ITEM WIDTH> 35
    ITEM OUTPUT WIDTH> 35
    ITEM TYPE> c
    ITEM NAME> symbol
    ITEM WIDTH> 3
    ITEM OUTPUT WIDTH> 3
    ITEM TYPE> i
```

This creates the .lut template. To add data to the info file:

```
ENTER COMMAND> ADD
    LTYPE> fault_ball
    SYMBOL> 16
```

- To create a lookup table in INFO for a **shadeset** (polygon unit names):

NOTE: This file must be created for each new polygon coverage (geologic map).

INFO:

```
ENTER COMMAND> DEFINE NAME_POLY.LUT
ITEM NAME> ptype
ITEM WIDTH> 5
ITEM OUTPUT WIDTH> 5
ITEM TYPE> c
ITEM NAME> symbol
ITEM WIDTH> 3
ITEM OUTPUT WIDTH> 3
ITEM TYPE> i
```

This creates the .lut template.

Data may be added to the INFO file by either:

- Entering the data directly:

INFO:

```
ENTER COMMAND> SELECT NAME_POLY.LUT
ENTER COMMAND> ADD
PTYPE> Qa
SYMBOL> 72
```

- Adding from an ASCII file:

The ASCII file must be comma delimited with the item values in the same order as the item names in the INFO file.

EXAMPLE:

```
Qa,72
Tfg,415
Kbs,870
Mr,353
Cm,478
Xlg,65
```

INFO:

```
ENTER COMMAND> SELECT NAME_POLY.LUT
ENTER COMMAND> ADD FROM /PATHNAME/LUT.TXT
ENTER COMMAND> SORT ON PTYPE
```

NOTE: Lookup-tables must be sorted on the relate item. Blank fields will default to the first symbol in the lookup table or to symbol 14. Therefore, include 'xx' as a placeholder in the markerset and shadeset lookup tables and set the symbol value to 0. Calc all blanks to 'xx':

```
AE: sel ptype = ''
AE: calc ptype = 'xx'
```

c. Relate files

A relate file is used to connect two INFO tables that share a common attribute. We use the same relate file in each workspace to temporarily associate descriptive information for a feature (e.g., arc, polygon, or point) and a specific symbol in the lookup table by means of a common item.

NOTE: The relate file geo.rel is already created on hydra. Use the newlut.aml to copy the relate file, geo.rel, and geolin.lut and geomrk.lut to a new workspace (ATTACHMENT 5).

- 1) Establish the relate environment:

Arc: **&r newlut.aml**

The following commands are listed below for information only

AE: relate add	
Relate Name: arcrel	(no more than eight characters)
Table Identifier: geolin.lut	(relates ltype item in .aat to ltype in lineset)
Database Name: info	
INFO item: ltype	
Relate Column: ltype	
Relate Type: ordered	
Relate Access: rw	
Relate Name: dipptrel	(relates pttype item in .pat to pttype in markerset)
Table Identifier: geomrk.lut	
Database Name: info	
INFO item: pttype	
Relate Column: pttype	
Relate Type: ordered	
Relate Access: rw	
(HIT RETURN TO END)	
AE: relate save <name>.rel	(saves relate parameters to an info file)

- 2) Apply the relate environment to the current Arcedit session:

AE: relate restore <name>.rel	
AE: symbolitem arc arcrel//symbol	(draws arcs with assoc. line symbol in lineset)

- 3) Add to the relate environment if needed:

AE: relate restore <name>.rel	
AE: relate add	(will prompt as above)

- 4) To list the current relate environment:

AE: relate list

d. Adding items to feature attribute tables

Feature attribute tables are INFO files containing descriptive information about the spatial data. These tabular INFO data files consist of items and records. An item stores categorical attribute information for all features in a database, while a record stores all information about one occurrence of a feature. Standard items are generated in a feature attribute table for each type of feature when the digital map or coverage is built. In

Relationships between features and their symbols are set up as previously described using the relate files, which must be restored at the beginning of each Arcedit session in order to be effective (set in startup aml).

When the arcs are fully edited and attributed, the coverage is built for polygons. The pulldown poly menu is customized to include all unit names (ptype) and is then used to tag all the polygons. When problems are encountered, such as incompletely closed polygons, the user can move easily back into the arc pulldown menu to correct the arcs. The coverage will then have to be built again for polygons in order to recreate polygon topology. When moving between menus for arc editing and tagging and polygon editing and tagging, it is critical that the coverage be built with the nodups nodiffs option in Arcedit to establish topology. Otherwise polygon attribute data can be lost. Once the coverage is completely digitized and edited a plot is produced and overlain on the source map to be checked for accuracy.

The next section consists of hints and suggestions for using startup environments and menus.

NOTE: At the beginning of an edit day, create a backup cover by copying the current coverage at the arc prompt before invoking the menus, as follows:

Arc: copy <cover_name> <cover_name.date>

In addition, archive the backup files regularly to tape or zip disks.

2. Startup AMLs

Generic pull-down menus (a series of nested AML files that simplify editing) are used for editing linework and annotation, and custom menus are used for labeling polygons (described in detail later). The editarc and editpoly startup amls (**ATTACHMENT 7**) are used to set up the editing environment and invoke the arcedit and polyedit pulldown menus. The editarc.aml is invoked during sessions when the arcs are being edited. Once polygon topology is created by building the coverage for polygons, the menus can be accessed by either the arc or polygon startup aml, depending on the purpose of the editing session.

For editing arcs:

Arc: &r editarc.aml

For editing polygons after topology is built:

Arc: &r editpoly.aml

The startup amls were written in vi editor and are designed to set up the working environment for editing coverages and for starting the associated nested menus. Within the AMLs the following variables may be changed to suit the users' needs.

NOTE: Most of these commands are also menu choices and may be changed at any time during the editing process using a pull-down option or by opening a command line and typing in the desired commands.

a. **Display:** These commands alter the way in which the coverage is displayed during the edit session:

- **setenv CANVASCOLOR <color>**

Choose either white or black for the background color of the graphics windows in your arc session.
Set this up in your .cshrc file in your home directory to suit your preference.

- **display 9999 {option}**

Sets the graphics display device and positions the display window. Follow the 9999 (x-terminal) with a 1, 2, 3, or 4 for preset graphics window sizes. Clicking and dragging the corners can also

be used to change the size of the window. There are additional arguments for this command, check ARC/INFO help.

- **edit <cover_name>**
The coverage you will be editing.
- **mape {bnd | tic} <cover_name>**
Map extent or geographic area of the coverage you are editing or of the view you need. For instance, if tics lie outside of the polygons the tic option ensures that they will be included in the map extent and not be graphically clipped. You can set multiple map extents by specifying each coverage's name after the mape command:
mape <name_cover1> <name_cover2>
- **drawenvironment {cover_name} <arc | tic ids | node errors>**
Specifies the features to be drawn on screen for the edit coverage. To display dangling nodes and pseudo nodes in color give the following commands:
nodecolor dangle {color}
nodecolor pseudo {color}
For additional arguments check ARC/INFO help.
- **bc <cover_name> {symbol}**
Specifies a coverage(s) to be used as a background display. For instance, when adding dip annotation it is useful to have the poly layer set in the background with the unit annotation and arcmarkers displayed so that the dip annotation doesn't overlap other map information. A symbol number from the lineset may be used to show backcoverages in a color to readily distinguish them from the edit coverage.
- **backenvironment {backcover_name} <arc | node | label | tic | anno | polygon>**
Specifies the features to be drawn on screen for the backcoverage. For additional argument check ARC/INFO help.
- **setdrawsymbol 0 <color>**
Changes the color of the selected features from the default <yellow> to your preference.
- **image <image>**
Specifies an image to be used as a background display. Useful for displaying TM images and scanned grids in the background for placement purposes. There are many color formatting options for a background image, check ARC/INFO help.

- b. **Scale:** During an edit session it is useful to display coverages exactly as they will be plotted in order to correctly size and place symbolized features. Use the following commands to set up the environment:

- **pagesize <width height>**
Sets the size of the paper on which your map will be plotted.
- **pageunits <inches | cm | pageunits_per_inch>**
Sets the units used to measure the pagesize (we usually use inches which is the default).
- **mapunits <inches | ft | cm | meters | mapunits_per_inch>**
Sets the coverage units of your map according to the projection (generally meters).
- **mapscale <scale>**
Sets the scale of your map. Use the denominator only, i.e., mapscale 100000 sets the scale to 1:100,000.
- To adjust the scale of marker symbols for a variety of mapscales do the following:
markerscale <scale>
Where <scale> is an absolute scaling factor greater than 0. The default scale is 1, therefore, a scale less than 1 will decrease the size of the marker and a scale greater than 1 will increase the size of the marker.

markerscale mapscale <scale>

Where <scale> is the scale at which symbols are accurately sized.

- c. **Symbolizing features:** To display attributed features with the correct symbol include these commands in your startup AML:

- **relate restore <info_file>**

Appy the established relate environment to the current editing session. This relates the symbol item in the feature attribute table to the lookup table (.lut).

- **symbolitem <cover_name> <feature> <relate_name>//<item>**

This command specifies the item from the relate file used to assign symbols to the feature class in the coverage. Once the relate is restored and this command given, the features will be drawn using the assigned symbol. This command must be preceded by a command specifying the symbolset, i.e., **lineset geo.lin** for feature <arc> and **markerset geo.mrk** for feature <point>.

- d. **Editing tolerances:** Tolerance refers to the amount of coordinate movement allowed, as well as the number of vertices on an arc and the distance between them as you digitize. Settings should vary with the map scale and type of feature being digitized. In addition to scale, effective tolerances are contingent on the straightness or curve of an arc and may need to be altered accordingly. Typically, man-made features require fewer vertices than natural features because they are straighter and more angular. Although there are suggested tolerances, experimentation is the best way to determine what is best for the current coverage and feature.

- **weedtolerance {distance | *}**

Specifies the minimum distance between vertices on an added arc. Weedtolerance is used to delete excessive vertices along a straight arc.

- **grain {distance | *}**

Specifies the minimum distance between vertices along an arc. Grain is used to spline curved arcs.

- **fuzzy tolerance**

Specifies the minimum distance separating nodes and vertices, as well as the distance a coordinate can move when creating topology. Fuzzy tolerance is specified as an argument in many ARC/INFO commands. Since coordinates within the fuzzy tolerance of each other are considered equal, coordinates may be moved to other locations. This effects the way in which your coverage is processed during the ‘clean’ command. To prevent fuzzy creep try to avoid cleaning coverages by identifying and correcting arc intersect errors as they occur.

- e. **Snapping tolerances:** Snapping environments, specifying when and how one feature will snap to another, are also dependent on the map scale and editing goal. An appropriate snapping environment can help eliminate coverage errors such as undershoots and overshoots. Overall, if snapping tolerances are too large for the data and/or the vertices are too far apart, the wrong nodes or arcs may snap together. Similarly, snapping will not occur if the tolerances are too short. Tolerances, therefore, must be monitored and manually adjusted to fit the data.

- **sc <cover_name>**

Sets the snapping coverage. Usually set to the edit coverage, but can be used to snap features in one coverage to features in another coverage. For instance, pointmarkers such as lineation symbols can be snapped to their associated arcs in another coverage layer.

- **sf <feature to snap> <feature to snap to>**

Controls which features snap together (ex: **sf node arc**).

- **snapping <first | closest> {distance | *}**
Sets the search radius for snapping features together. Experiment with distances as effectiveness varies with mapscale. The user can interactively set the snapping distance on-screen by using *. Generally, if digitizing in map units, the smaller the scale the larger these distances.
- **nodesnap <first | closest> {distance | *}**
Sets the search radius for snapping nodes together.
- **arcsnap on <* | distance>**
Controls whether the ends of arcs will snap to existing arcs or not.
- **snaporder node <vertex | segment>**
snaporder vertex <node | segment>
snaporder segment <node | vertex>
Controls the search order for snapping features to arcs.

f. **Determining tolerances:** Tolerances are either set explicitly, by entering a value, or interactively using either the digitizer cursor or the mouse, by specifying * as the tolerance distance. Any tolerance value may be easily changed by using the ‘tol’ pulldown on the arcedit.menu, which allows for user determined distances. Again, experimentation is the best way to decide upon appropriate tolerances

Set tolerances in meters when working in a projected coverage where meter is the map unit. Upon saving a coverage, the tolerances in use at the time are saved in a .tol file, which is opened each time you open the coverage..

The following recommended tolerances are based on ESRI suggestions and digitizer resolution and are set as defaults in the editarc and editpoly AMLs.:

<u>Map scale</u>	<u>.002 fuzzy tolerance</u>	<u>.02 in. weed and grain</u>	<u>.05 in. snapping</u>
1:24,000	1.219 m	12.192 m	30.480 m
1:48,000	2.784 m	24.384 m	60.960 m
1:62,500	3.175 m	31.750 m	80.465 m
1:100,000	5.08 m	50.80 m	127 m
1:250,000	12.70 m	127 m	317.50 m

g. **Other useful commands for edit sessions:**

- **intersectarcs <off | add | all>**
Specifies whether or not arc intersections will be calculated and nodes added during editing. Setting intersectarcs on limits the number of intersection errors that must be addressed before a coverage can be built for topology.
- **duplicatearcs <yes | no>**
Specifies whether or not an arc can be added with the same starting and ending nodes as an existing arc. Set this parameter to prevent the same arc from being digitized twice; the error will be reported and the arc deleted. However, there are often legitimate occurrences of two arcs sharing the same nodes, so be careful.

3. Pulldown menus

Menus are a graphical organization of ARC/INFO commands that are nested for convenience and efficiency, allowing the user multitasking capabilities when editing. The menus we have created are invoked through startup AMLs. Menus may be written and altered to suit personal preferences, using a text editor such as **vi**. They are restricted to 20 headers, so it is best to keep the menus streamlined to the task at hand. The polyedit and arcedit menus are nested to allow switching between editing and tagging arcs and tagging polygons with the proper unit names (**ATTACHMENT 8**). The attitude and annoedit menus (described in later sections) are nested to facilitate placement of annotation. Some common preferences and suggestions follow:

- **Feature attributes:** following the ltype and ltype choices you may wish to add the command draw nocalar to display symbolized features immediately.
- **Editing:** Add commands suited to your map such as **merge**, useful in eliminating sliver polygons.
- **New:** This command allows you to choose the ltype you work on so that is will be automatically tagged as you digitize. Choose **new** on the menu, then the ltype, and continue tagging until you are ready for a new ltype.
- **Command pulldown:** With this option, at any time you can run other amls and change parameters not accommodated by the menu.

4. Editing process

This section discusses the process of editing a coverage in ARCEDIT and provides hints and tips. New users should refer to ARC/INFO documentation for a more complete description.

- **Moving around the graphics screen:**
 - a. **Control e** – extent; allows the user to zoom in to the chosen part of the screen
 - b. **Control v** – zoom center
 - c. **Control f** – fullview
 - d. **Control a** – pan around the image
 - e. **Control e** – opens a new window allowing several map views at once. It is useful to have a zoom window and an overview window simultaneously.
- **Digitizing and the RMS Error:** A Root Mean Square error is calculated for the difference between tics entered for registering a digitizing session and the actual tic coordinates in the coverage. The pulldown command **coordinate dig <cover_name>** prompts you to enter four tic marks from the map.. A large or complicated map may require more than one digitizing session. When your map is placed on the digitizing table and you need more than one session to complete the digitizing process, you may have to remove the map from the table. Upon completion ARC calculates the Root Mean Square Error, given in both digitizer units and map units. In a typical session, you want the digitizer units to be .004 or less and no more than .008. The value will vary somewhat with the mapscale and with the quality of the base material. Once the RMSE is established in the first digitizing session, it is best to try and attain the same value in all subsequent sessions.
- **Coordinate digitizer, coordinate mouse:** These pulldown commands allow the user to toggle between digitizer table input and screen input. Digitizing can be tedious; by using the **coordinate mouse** command the

user can edit digitized lines in order to break up long digitizing sessions. To restart digitizing after editing, give the **coordinate digitizer** command to resume digitizing.

- **Digitizing errors:** During a digitizing session, pay attention to the graphics screen and xterm window. Digitizing options are listed, and features are drawn to the screen as they are digitized; errors can be most easily corrected at this stage. For instance, if you make an error in the line, pressing the #4 key on the digitizing puck will remove the last vertex added. By reading the messages on the screen, you can eliminate errors and reduce editing time.
- **Saving and building:** Save your work frequently using the **save** command. Doing so prevents redoing lost work in the event of a system crash or other mishap. Build the cover from the command pulldown or from the ARC prompt or ARCEDIT prompt. The default for build is polygon, so if working with a point coverage, be sure to specify **build <cover_name> point**.
- One common problem in polygon editing are sliver polygons and unlabelled polygons. One method to correct these is to use the **tiptoe.aml** command (**ATTACHMENT 9**):

```
AE: ef poly  
AE: sel ptype = ''  
AE: tiptoe init
```

You can then edit, merge, dissolve, tag, or delete each poly. To continue after editing a polygon:

```
AE: tiptoe next  
If you exit the tiptoe aml, you can pick up where you left off:  
AE: tiptoe reinit
```

E. Structure line layer <name_fold>

1. Scan

If working from a scan, these features will have been already attributed in <name_poly>. Therefore, move folds, dikes, etc from <name_poly> into this cover. If digitizing, create this cover and add (digitize) these features to the new coverage to avoid interference with unit polygons.

*NOTE: The additemfold.aml creates the coverage <name_fold> and adds items to the .aat.
(ATTACHMENT 10).*

Arc: &r additemfold.aml

The following commands are included in the aml and are listed below for information only.

Arc: create <name_fold> <name_poly>

Add custom items to .aat.

```
Arc: additem <name_fold.aat> <name_fold.aat> ltype 35 35 c  
Arc: additem <name_fold.aat> <name_fold.aat> pttype 35 35 c  
Arc: additem <name_fold.aat> <name_fold.aat> plunge 3 3 i
```

Move structures from <name>_poly.

```
AE: ec <name_poly>  
AE: ef arc  
AE: sel ltype = "dike"  
AE: put <name_fold>
```

Continue for all folds, dikes, etc.

Arc: build <name_fold> line

2. Digitized cover

If digitizing structures in <name_fold>, proceed as with creating <name_poly>, using arcedit.menu

NOTE: The item ‘plunge’ stores an integer that determines whether or not an arrowhead is drawn using the AP command ‘arcarrows’:

0 specifies no arrowhead

A positive value draws an arrowhead at the to-node

A negative value draws the arrowhead at the from-node

F. Structure point layer <name_dip>

1. Adding items

Creation of the <name_dip> and adding of items is done with the **additemdip.aml**.
(ATTACHMENT 11).

Arc: &r additemdip.aml

The following commands are included in the aml and are listed below for information only.

Arc: create <name_dip> <name>_poly
Create point attribute table and add custom items
AE: ec <name_dip>
AE: ef point
AE: createattributes
AE: additem ptype 35 35 c
AE: additem dip 3 3 i
AE: additem strike 3 3 I
AE: ef arc
AE: createattributes (the AML uses an arc to calculate strike and marker placement)

2. Adding attitudes

To digitize attitudes and other structural point markers (ptype), we use **attitude.aml**, which invokes attitude.menu and uses the plunge and dip amls written by Haydee Hampton (ATTACHMENT 12).

Arc: &r attitude.aml

Hints for using the menu:

First, set snapping using the ‘snapping – set snapping’ pulldown field.
For digitizing, use ‘COORDINATE – enable dig’ then ‘dig cover’.
To snap lineation to foliation, use the ‘snapping - snap to edit cover’ pulldown.
To snap fault dip symbols to faults, use the ‘snapping – snap to back cover’ pulldown.

NOTE: The coordinate device can be changed on dip.aml and plunge.aml to use the mouse or digitizer for entering data.

The following commands are included in the attitude.menu and are listed below for information only.

To set up dip annotation (do this after all attitudes are digitized):

AE: ef anno []
AE: annofeature point dip
AE: annocapture
AE: calc \$size = 200 (for 1:100,000 scale)
= 120 (for 1:62,500 scale)
= 50 (for 1:24,000 scale)
AE: save

If new attitudes are added to <name_dip> after creating the dip annotation (step 5 above), the associated new annotations must be added, using commands below:

AE: ef anno
AE: annofeature point dip
AE: add new
AE: annoselectfeature * [sel point for which need to add new annotation]
AE: annotext feature
AE: annoplace feature

G. Annotation layer <name_anno>

1. Creating annotation

Annotation is a cartographic layer for placing unit labels and other text. Creation of the annotation layer is done by invoking the ***createanno.aml*** (ATTACHMENT 13).

Arc: &r createanno.aml

The following commands are included in the aml and are listed below for information only.

Arc: copy <name_poly> <name>_anno

Capture the unit names held in the feature attribute table (pat).

AP: mape <name_anno>

AP: annocoverage <name_anno> unit (creates annotation subclass)

AP: textset font.txt

AP: textsymbol 1 /*Univers Medium

AP: labeltext <name_anno> ptype

Recalculate text size in mapunits (be sure \$fit is set to off)

AE: mapunits meters

AE: mapscale 24000

AE: ef anno.unit

AE: calc \$size = 60

for 1:24,000, use 60

for 1:62,500, use 180

for 1:100,000, use 220

Add ltype to .aat for leaders:

AE: additem ltype

2. Manipulating annotation

Annotation should be placed to avoid overlap with other symbols. We manipulate placement of the dip annotation as much as possible prior to creating and moving the annotation text. To move annotation text, use the **announitedit.aml**, which invokes the **annoedit.menu** (ATTACHMENT 14). Leaders for annotation text not placed within the appropriate polygon are added through this menu. Dip annotation can also be manipulated through nesting of the **attitude.menu** into the annotation menu.

Arc: &r announitedit.aml

The following commands are included in the aml and are listed below for information only.

Set drawenvironment to edit the annotation coverage

AE: mape <name_anno>

AE: ec <name_anno>

AE: mapunits meters

AE: mapscale 24000
AE: lineset geo.lin
AE: textset font
AE: markerset geo.mrk
AE: de arc anno.unit
AE: ef anno.unit

Set additional backcoverages to avoid repositioning the annotation on another feature.

AE: bc <name_poly> <color or symbol number from lineset>
AE: bc <name_dip> <color or symbol number from lineset>
AE: be <name_poly> arc
AE: be <name_dip> point anno

Use the drag or move command rather than the annoplace command - annoplace will change the shape of the text string.

Arcedit: sel
Arcedit: drag

To view and zoom at 24000 scale. Pan to move around on the graphic page

AE: pageextent (define box for zoom)
AE: draw (must follow pageextent command with draw)
AE: pageextent page (returns to fullview)

Leaders (arcs) are also drawn in this layer. Must add ltype to .aat.

III. TABLE SUMMARIZING DIGITAL STRUCTURE OF GEOLOGY COVERAGES

Coverage	Attribute table	User-defined Items	Usage
<name_poly>	<name_poly.pat>	PTYPE 5 5 C PATTERN 3 3 I ORIGPTYPE 5 5 C	Ptype is used as relate item in <name_poly>.lut which relates ptype (unit name) to shadeset
	<name_poly.aat>	LTYPE 35 35 C PTTYPE 35 35 C	Ltype is relate item in geolin.lut which relates arcs to line symbol Pttpe is relate item in geomrk.lut which relates a marker symbol (such as fault ball) to an arc (arcmarker)
<name_fold>	<name_fold.aat>	LTYPE 35 35 C PTTYPE 35 35 C PLUNGE 3 3 I	Ltype is relate item in geolin.lut which relates arc to line symbol (fold axis) Pttpe is relate item in geomrk.lut which relates a fold marker symbol to an arc (fold axis)(arcmarker) Plunge stores degree of plunge of fold axes
<name_dip>	<name_dip.aat>	none	Used only for calculating strike when digitizing
	<name_dip.pat>	PTTYPE 35 35 C DIP 3 3 I STRIKE 3 3 I PLUNGE 3 3 I	Pttpe is relate item in geomrk.lut which relates a marker symbol (such as strike and dip symbol) to a point Dip item stores attitude dip Strike item stores azimuthal strike of attitude Plunge items stores degree of plunge of lineation
<name_anno>	<name_anno.pat>	None (all annotation information stored as pseudoitems (\$all))	Carryover from creating annotation layer from <name_poly>
	<name_anno.aat>	ltype	Used only for storing annotation leaders

ATTACHMENT 1

LOOKUP TABLE FOR GEO.MRK (geomrk.lut)

Record	PTTYPE	SYMBOL
1	air_foliation	17
2	anticline	52
3	antiform_axis	74
4	approx_bedding	4
5	bedding	1
6	bedding_tops	8
7	box	68
8	box_fill	69
9	circle	70
10	circle_fill	71
11	clay_pit	76
12	crumpled_bedding	5
13	crumpled_foliation	18
14	diamond	74
15	diamond_fill	75
16	dike	90
17	dike_box	92
18	dike_x	94
19	double_arrow	64
20	fault_ball_fill	42
21	fault_ball_fill_lg	43
22	fault_ball_open	44
23	fault_dip	63
24	fault_dip_fill	41
25	fault_dip_open	40
26	fault_plane	37
27	fault_ss_ll	39
28	fault_ss_ll_bar	67
29	fault_ss_rl	38
30	fault_ss_rl_bar	66
31	fold_axis_fill	46
32	fold_axis_open	45
33	fold_axis_plunge	49
34	foliation	14
35	foliation_and_bedding	13
36	gravel_pit	76
37	horizontal_bedding	3
38	horizontal_dike	93
39	horizontal_foliation	16
40	horizontal_joint	28
41	horizontal_line	86
42	horizontal_lineation	58
43	horizontal_vein	82
44	horizontal_vein_dot	85
45	igneous_air_foliation	23
46	igneous_crumpled_foliation	24
47	igneous_foliation	20
48	igneous_horizontal_foliation	22
49	igneous_vertical_foliation	21

50	incipient_crenulation_clvg	33
51	inclined_clvg	34
52	inclined_clvg_tops	35
53	joint	26
54	lineation	60
55	magmatic_foliation	20
56	magmatic_horizontal_foliation	22
57	magmatic_vertical_foliation	21
58	main_crenulation_clvg	32
59	metamorphic_foliation	14
60	metamorphic_horizontal_foliation	16
61	metamorphic_vertical_foliation	15
62	mine	77
63	mineral_lineation	60
64	minor_fold	88
65	monocline	56
66	mylonitic_foliation	34
67	mylonitic_lineation	61
68	ot_anticline	50
69	ot_bedding	6
70	ot_bedding_tops	7
71	ot_syncline	51
72	pink_anticline	11
73	pink_monocline	25
74	pink_ot_anticline	86
75	pink_ot_syncline	87
76	pink_syncline	12
77	quarry	77
78	relict_graded_bedding	89
79	relict_pillows	59
80	sfold	48
81	shaft_inclined	79
82	shaft_vertical	78
83	shear_sense	10
84	slickenlines_fill	19
85	slickenlines_open	65
86	small_anticline	54
87	small_syncline	55
88	syncline	53
89	synform_axis	36
90	triangle	72
91	triangle_fill	73
92	unmineralized_horizontal_joint	31
93	unmineralized_joint	29
94	unmineralized_vertical_joint	30
95	vein	80
96	vein_dot	83
97	vertical_bedding	2
98	vertical_bedding_tops	9
99	vertical_dike	91
100	vertical_dike_x	95
101	vertical_foliation	15
102	vertical_joint	27
103	vertical_lineation	57
104	vertical_vein	81
105	vertical_vein_dot	84
106	volcanic_vent	62

107	xpicks	77
108	xx	0
109	zfold	47

Scale (X,Y) = (1223.170,1223.114) Skew (degrees) = (-0.007)
 Rotation (degrees) = (-1.680) Translation = (223716.061,3983315.663)
 RMS Error (input,output) = (0.003,3.079)

Affine X = Ax + By + C

Y = Dx + Ey + F

A =	1222.645	B =	35.706	C =	223716.061
D =	-35.852	E =	1222.593	F =	3983315.663

tic id	input x output x	input y output y	x error	y error
1	16.964 245138.342	19.013 4005954.472	-2.462	-1.845
2	7.769 233891.480	19.021 4006290.217	2.466	1.848
3	7.770 234032.529	22.799 4010912.829	-2.465	-1.848
4	16.968 245273.417	22.797 4010576.929	2.461	1.845

EDITING:

Added items for attributing:

to .aat - ltype, ptype, symbol
 to .pat - ptype, symbol

Digitized with menu in UTM coordinates and attributed linework in the process.
 Snapping tolerances set at 40 meters; grain and weed at 10 meters. Registration
 RMS error maintained at .002 inches.

ATTACHMENT 5

```
/* NEWLUT.AML
/* AML adds lookup tables and relate file for lineset and markerset
/* - geo.lin & geo.mrk

&if [exists geolin.lut -info] &then
&do
  &type [delete info!arc!geolin.lut -info]
  copyinfo /arcwork2/geolin.lut
  &type !old geolin.lut deleted and replaced!
&end

&else &do
  copyinfo /arcwork2/geolin.lut
  &type !geolin.lut added to directory!
&end

&if [exists geomrk.lut -info] &then
&do
  &type [delete info!arc!geomrk.lut -info]
  copyinfo /arcwork2/geomrk.lut
  &type !old geomrk.lut deleted and replaced!
&end

&else &do
  copyinfo /arcwork2/geomrk.lut
  &type !geomrk.lut added to directory!
&end

&if [exists geo.rel -info] &then
&do
  &type [delete info!arc!geo.rel -info]
  relate add
  arcrel
  geolin.lut
  info
  ltype
  ltype
  ordered
  rw
  dipptrel
  geomrk.lut
  info
  ptttype
  ptttype
  ordered
  rw
  [unq ' ']
  relate save geo.rel
  &type !old geo.rel deleted and replaced!
&end

&else &do
  relate add
  arcrel
  geolin.lut
  info
  ltype
  ltype
  ordered
  rw
  dipptrel
  geomrk.lut
  info
  ptttype
  ptttype
  ordered
  rw
```

```
[unq ' ']
relate save geo.rel
&type !geo.rel added to directory!
&end

&return
```

ATTACHMENT 6

```
/* ADDITEMPOLY.AML
/* AML adds items to .aat and .pat of selected coverage.

&if [EXISTS geolin.lut -info] AND ~
[EXISTS geomrk.lut -info] &then

&do
AE
&sv cover = [GETCOVER * -all 'Select a coverage']
MAPE %cover%
EC %cover%

EF arc
&if [EXISTS %cover% -arc] &then
&do
    ADDITEM ltype 35 35 c
    ADDITEM pttype 35 35 c
    ADDITEM symbol 3 3 i
    &end
&else
&do
    CREATEATTRIBUTES
    ADDITEM ltype 35 35 c
    ADDITEM pttype 35 35 c
    ADDITEM symbol 3 3 i
    &end

BUILD NODUPS NODIFFS
EF poly
ADDITEM pttype 5 5 c
ADDITEM pattern 3 3 i
ADDITEM origpttype 5 5 c
save
quit
&end

&else
&type !geolin.lut and geomrk.lut must be in INFO directory! RUN newlut.aml!

&return
```

ATTACHMENT 7

```
/* EDITARC.AML
/* AML to invoke menu for input and editing linework.
/* Menu includes coordinate change for digitizing.

display 9999 size 1280 650 pos 100 75
ae

&sv .cover = [getcover * -arc 'Select a coverage']
mape %.cover%
mapunits meters
&sv .mapscale = [response 'Enter the mapscale']
mapscale %.mapscale%
ec %.cover%

lineset geo.lin
relate restore geo.rel
symbolitem arc arcrel//symbol

sc %.cover%
&sv .snaptol = %.mapscale% * .0254 * .05
snapping closest %.snaptol%
arcsnap on %.snaptol%
nodesnap closest %.snaptol%
sf node arc
snaporder segment node
&sv .weedtol = %.mapscale% * .0254 * .02
weedtolerance %.weedtol%
&sv .graintol = %.mapscale% * .0254 * .02
grain %.graintol%

editdistance default
setdrawsymbol 0 magenta
drawselect all
intersectarcs all
weeddraw off

de arc tic id
ef arc
draw

&terminal 9999
&fullscreen &popup
&menu arcedit.menu &stripe arcedit.menu

&return

/* EDITPOLY.AML
/* Setup aml to invoke menu for tagging polygons

display 9999 size 1280 650 pos 100 75
ae

&sv .cover = [getcover * -poly]
mape %.cover%
mapunits meters
&sv .mapscale = [response 'Enter the mapscale']
mapscale %.mapscale%
ec %.cover%

lineset geo.lin
relate restore geo.rel
symbolitem arc arcrel//symbol

sc %.cover%
&sv .snaptol = %.mapscale% * .0254 * .05
snapping closest %.snaptol%
```

```
arcsnap on %.snaptol%
nodesnap closest %.snaptol%
sf node arc
snaporder segment node
&sv .weedtol = %.mapscale% * .0254 * .02
weedtolerance %.weedtol%
&sv .graintol = %.mapscale% * .0254 * .02
grain %.graintol%

editdistance default
setdrawsymbol 0 magenta
drawselect all
intersectarcs all
weeddraw off

de arc tic
ef poly
draw

&terminal 9999
&fullscreen &popup
&menu polyedit.menu &stripe polyedit.menu

&return
```

ATTACHMENT 8

```

1 arcedit.menu
TOL
'Set snapcover'
    &sv snpcov = [getcover * -all 'Select snapcover']; ~
    sc %snpcov%
'Set snapfeatures'
    &sv.snpfeat1 = [getchoice 'arc' 'node' 'point' ~
        -prompt 'Select edit feature']; ~
    &sv.snpfeat2 = [getchoice 'arc' 'node' 'point' ~
        -prompt 'Select snap feature']; ~
    sf %snpfeat1% %snpfeat2%
'Set snaporder'
    &sv.snpord1 = [getchoice 'node' 'vertex' 'segment' ~
        -prompt 'Select first feature']; ~
    &sv.snpord2 = [getchoice 'node' 'vertex' 'segment' ~
        -prompt 'Select second feature']; ~
    snaporder %snpord1% %snpord2%
    snapping closest *
    weedtolerance *
    grain *
    status tolerances; status snap
    &sv command = [response 'Type Command']; ~
    &lv command; [unq %command%]

'Flt NORMAL'
'normal flt certain'
moveitem 'normal_flt_certain' to ltype; calc pttype = 'xx'; ~
&type normal_flt_certain; calc symbol = 2; draw noclear
'mormal flt approx'
moveitem 'normal_flt_approx' to ltype; calc pttype = 'xx'; ~
&type normal_flt_approx; calc symbol = 2; draw noclear
'mormal flt infer'
moveitem 'normal_flt_inferred' to ltype; calc pttype = 'xx'; ~
&type normal_flt_inferred; calc symbol = 2; draw noclear
'mormal flt infer ?'
moveitem 'normal_flt_inferred_queried' to ltype; ~
calc pttype = 'xx' ~
&type normal_flt_inferred_queried; calc symbol = 2; draw noclear
'mormal flt concealed'
moveitem 'normal_flt_concealed' to ltype; calc pttype = 'xx'; ~
&type normal_flt_concealed; calc symbol = 2; draw noclear
'mormal flt conceal ?'
moveitem 'normal_flt_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type normal_flt_concealed_queried; calc symbol = 5; draw

noclear
'high angle certain'
moveitem 'high_angle_flt_certain' to ltype; calc pttype = 'xx'; ~
&type high_angle_flt_certain; calc symbol = 5; draw noclear
'high angle approx'
moveitem 'high_angle_flt_approx' to ltype; calc pttype = 'xx'; ~
&type high_angle_flt_approx; calc symbol = 5; draw noclear
'high angle concealed'
moveitem 'high_angle_flt_concealed' to ltype; ~
calc pttype = 'xx'; ~
&type high_angle_flt_concealed; calc symbol = 5; draw noclear
'low angle normal certain'
moveitem 'low_angle_norm_flt_certain' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_certain; calc symbol = 4; draw noclear
'low angle normal approx'
moveitem 'low_angle_norm_flt_approx' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_approx; calc symbol = 4; draw noclear
'low angle normal infer'
moveitem 'low_angle_norm_flt_inferred' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_inferred; calc symbol = 4; draw noclear
'low angle normal infer ?'
moveitem 'low_angle_norm_flt_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_inferred_queried; calc symbol = 4; ~
draw noclear
'low angle normal concealed'
moveitem 'low_angle_norm_flt_concealed' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_concealed; calc symbol = 4; draw

noclear
'low angle normal concealed ?'
moveitem 'low_angle_norm_flt_concealed_query' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_norm_flt_concealed_query; calc symbol = 4; ~
draw noclear
'low angle normal solid'
moveitem 'low_angle_normal_flt_solid' to ltype; ~
calc pttype = 'xx'; ~
&type low_angle_normal_flt_solid; calc symbol = 4; draw noclear
'fault certain'
moveitem 'fault_certain' to ltype; calc pttype = 'xx'; ~

```

```

'fault approx'
'fault inferred'
'fault concealed'
'FLT OTHER'
'attenuation certain'
~
'attenuation approx'
~
'attenuation infer'
'attenuation infer ?'
'attenuation concealed'
'attenuation concealed ?'
'thrust certain'
'thrust approx'
'thrust infer'
'thrust infer ?'
noclear
'thrust concealed'
'thrust concealed ?'
noclear
'reverse certain'
'reverse approx'
'reverse infer'
'reverse infer ?'
noclear
'reverse concealed'
'reverse concealed ?'
noclear
'ss ll certain'
'ss ll approx'
'ss ll infer'
'ss ll infer ?'
'ss ll concealed'
&type fault_certain; calc symbol = 7; draw nclear
moveitem 'fault_approx' to ltype; calc pttype = 'xx'; ~
&type fault_approx; calc symbol = 7; draw nclear
moveitem 'fault_inferred' to ltype; calc pttype = 'xx'; ~
&type fault_inferred; calc symbol = 7; draw nclear
moveitem 'fault_concealed' to ltype; calc pttype = 'xx'; ~
&type fault_concealed; calc symbol = 7; draw nclear
moveitem 'attenuation_flt_certain' to ltype; calc pttype = 'xx';
&type attenuation_flt_certain; calc symbol = 9; draw nclear
moveitem 'attenuation_flt_approx' to ltype; calc pttype = 'xx';
&type attenuation_flt_approx; calc symbol = 9; draw nclear
moveitem 'attenuation_flt_inferred' to ltype; ~
calc pttype = 'xx'; ~
&type attenuation_flt_inferred; calc symbol = 9; draw nclear
moveitem 'attenuation_flt_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type attenuation_flt_inferred_queried; calc symbol = 9; ~
draw nclear
moveitem 'attenuation_flt_concealed' to ltype; ~
calc pttype = 'xx'; ~
&type attenuation_flt_concealed; calc symbol = 9; draw nclear
moveitem 'attenuation_flt_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type attenuation_flt_concealed_queried; calc symbol = 9; ~
draw nclear
moveitem 'thrust_flt_certain' to ltype; calc pttype = 'xx'; ~
&type thrust_flt_certain; calc symbol = 12; draw nclear
moveitem 'thrust_flt_approx' to ltype; calc pttype = 'xx'; ~
&type thrust_flt_approx; calc symbol = 12; draw nclear
moveitem 'thrust_flt_inferred' to ltype; calc pttype = 'xx'; ~
&type thrust_flt_inferred; calc symbol = 12; draw nclear
moveitem 'thrust_flt_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type thrust_flt_inferred_queried; calc symbol = 12; draw
moveitem 'thrust_flt_concealed' to ltype; calc pttype = 'xx'; ~
&type thrust_flt_concealed; calc symbol = 12; draw nclear
moveitem 'thrust_flt_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type thrust_flt_concealed_queried; calc symbol = 12; draw
moveitem 'reverse_flt_certain' to ltype; calc pttype = 'xx'; ~
&type reverse_flt_certain; calc symbol = 6; draw nclear
moveitem 'reverse_flt_approx' to ltype; calc pttype = 'xx'; ~
&type reverse_flt_approx; calc symbol = 6; draw nclear
moveitem 'reverse_flt_inferred' to ltype; calc pttype = 'xx'; ~
&type reverse_flt_inferred; calc symbol = 6; draw nclear
moveitem 'reverse_flt_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type reverse_flt_inferred_queried; calc symbol = 6; draw
moveitem 'reverse_flt_concealed' to ltype; calc pttype = 'xx'; ~
&type reverse_flt_concealed; calc symbol = 6; draw nclear
moveitem 'reverse_flt_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type reverse_flt_concealed_queried; calc symbol = 6; draw
moveitem 'ss_flt_ll_certain' to ltype; calc pttype = 'xx'; ~
&type ss_flt_ll_certain; calc symbol = 3; draw nclear
moveitem 'ss_flt_ll_approx' to ltype; calc pttype = 'xx'; ~
&type ss_flt_ll_approx; calc symbol = 3; draw nclear
moveitem 'ss_flt_ll_inferred' to ltype; calc pttype = 'xx'; ~
&type ss_flt_ll_inferred; calc symbol = 3; draw nclear
moveitem 'ss_flt_ll_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type ss_flt_ll_inferred_queried; calc symbol = 3; draw nclear
moveitem 'ss_flt_ll_concealed' to ltype; calc pttype = 'xx'; ~
&type ss_flt_ll_concealed; calc symbol = 3; draw nclear

```

```

'ss ll concealed ?'
moveitem 'ss_flt_ll_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type ss_flt_ll_concealed_queried; calc symbol = 3; draw noclear

'ss rl certain'
moveitem 'ss_flt_rl_certain' to ltype; calc pttype = 'xx'; ~
&type ss_flt_rl_certain; calc symbol = 3; draw noclear

'ss rl approx'
moveitem 'ss_flt_rl_approx' to ltype; calc pttype = 'xx'; ~
&type ss_flt_rl_approx; calc symbol = 3; draw noclear

'ss rl infer'
moveitem 'ss_flt_rl_inferred' to ltype; calc pttype = 'xx'; ~
&type ss_flt_rl_inferred; calc symbol = 3; draw noclear

'ss rl infer ?'
moveitem 'ss_flt_rl_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type ss_flt_rl_inferred_queried; calc symbol = 3; draw noclear

'ss rl concealed'
moveitem 'ss_flt_rl_concealed' to ltype; calc pttype = 'xx'; ~
&type ss_flt_rl_concealed; calc symbol = 3; draw noclear

'ss rl concealed ?'
moveitem 'ss_flt_rl_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type ss_flt_rl_concealed_queried; calc symbol = 3; draw noclear

CNTCT
'contact certain'
moveitem 'contact_certain' to ltype; calc pttype = 'xx'; ~
&type contact_certain; calc symbol = 15; draw noclear

'contact approx'
moveitem 'contact_approx' to ltype; calc pttype = 'xx'; ~
&type contact_approx; calc symbol = 14; draw noclear

'contact infer'
moveitem 'contact_inferred' to ltype; calc pttype = 'xx'; ~
&type contact_inferred; calc symbol = 14; draw noclear

'contact infer query'
moveitem 'contact_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type contact_infer_query; calc symbol = 14; draw noclear

'contact concealed'
moveitem 'contact_concealed' to ltype; calc pttype = 'xx'; ~
&type contact_concealed; calc symbol = 14; draw noclear

'contact concealed ?'
moveitem 'contact_concealed_queried' to ltype; ~
calc pttype = 'xx'; ~
&type contact_concealed_queried; calc symbol = 14; draw noclear

'contact indeterminate'
moveitem 'contact_ineterminate' to ltype; calc pttype = 'xx'; ~
&type contact_ineterminate; calc symbol = 14; draw noclear

MISC
'map bnd'
moveitem 'map_boundary' to ltype; calc pttype = 'xx'; ~
&type map_boundary; calc symbol = 8; draw noclear

'lake'
moveitem 'lake' to ltype; calc pttype = 'xx'; ~
&type lake; draw noclear

'dike'
moveitem 'dike' to ltype; calc pttype = 'xx'; ~
&type dike; draw noclear

'sag pond'
moveitem 'sag_pond' to ltype; calc pttype = 'xx'; ~
&type sag_pond; draw noclear

'sinkhole'
moveitem 'sinkhole' to ltype; calc pttype = 'xx'; ~
&type sinkhole; draw noclear

'landslide scarp'
moveitem 'landslide_scarp' to ltype; calc pttype = 'xx'; ~
&type landslide_scarp; draw noclear

'shear zone'
moveitem 'shear_zone' to ltype; calc pttype = 'xx'; ~
&type shear_zone; draw noclear

'pegmatite'
moveitem 'pegmatite' to ltype; calc pttype = 'xx'; ~
&type pegmatite; draw noclear

'conglomeratic marker'
moveitem 'conglomeratic_marker' to ltype; calc pttype = 'xx'; ~
&type conglomeratic_marker; draw noclear

'mylonite and cataclasite'
moveitem 'black_mylonite_and_cataclasite' to ltype; ~
calc pttype = 'xx'; ~
&type black_mylonite_and_cataclasite; draw noclear

FOLD
'anticline approx'
moveitem 'anticline_approx' to ltype; calc pttype = 'xx'; ~
&type anticline_approx; draw noclear

'anticline certain'
moveitem 'anticline_certain' to ltype; calc pttype = 'xx'; ~
&type anticline_certain; draw noclear

'anticline concealed'
moveitem 'anticline_concealed' to ltype; calc pttype = 'xx'; ~
&type anticline_concealed; draw noclear

'anticline inferred'
moveitem 'anticline_inferred' to ltype; calc pttype = 'xx'; ~
&type anticline_inferred; draw noclear

'anticline inferred queried'
moveitem 'anticline_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type anticline_inferred_queried; draw noclear

'antiform'
moveitem 'antiform' to ltype; calc pttype = 'xx'; ~
&type antiform; draw noclear

'syncline approx'
moveitem 'syncline_approx' to ltype; calc pttype = 'xx'; ~

```

```

'syncline certain'
'syncline concealed'
'syncline inferred'
'syncline inferred queried'

'synform'
'monocline certain'
'monocline concealed'
'ot anticline'
'ot anticline approx'
'ot syncline'
'ot syncline approx'
'plunging anticline'
'plunging anticline approx'

'plunging syncline'
'plunging syncline approx'

PTTYPE
'anticline'
'ot anticline'
'antiform axis'
'syncline'
'ot syncline'
'synform axis'
'fault ball fill'
'fault ball open'
'fault ss rl'
'fault ss rl bar'
'fault ss ll'
'fault ss ll bar'
'monocline'
'xx'

COORD
'Coordinate Mouse'
'Coordinate Digitizer'
'Dig Cover'
'Enable Altek'
'Run Digtest'
'Enable Summa'

&type syncline_approx; draw noclear
moveitem 'syncline_certain' to ltype; calc pttype = 'xx'; ~
&type syncline_certain; draw noclear
moveitem 'syncline_concealed' to ltype; calc pttype = 'xx'; ~
&type syncline_concealed; draw noclear
moveitem 'syncline_inferred' to ltype; calc pttype = 'xx'; ~
&type syncline_inferred; draw noclear
moveitem 'syncline_inferred_queried' to ltype; ~
calc pttype = 'xx'; ~
&type syncline_inferred_queried; draw noclear
moveitem 'synform' to ltype; calc pttype = 'xx'; ~
&type synform; draw noclear
moveitem 'monocline_certain' to ltype; calc pttype = 'xx'; ~
&type monocline_certain; draw noclear
moveitem 'monocline_concealed' to ltype; calc pttype = 'xx'; ~
&type monocline_concealed; draw noclear
moveitem 'ot_anticline' to ltype; calc pttype = 'xx'; ~
&type ot anticline; draw noclear
moveitem 'ot_anticline_approx' to ltype; calc pttype = 'xx'; ~
&type ot anticline approx; draw noclear
moveitem 'ot_syncline' to ltype; calc pttype = 'xx'; ~
&type ot syncline; draw noclear
moveitem 'ot_syncline_approx' to ltype; calc pttype = 'xx'; ~
&type ot syncline approx; draw noclear
moveitem 'plunging_anticline' to ltype; calc pttype = 'xx'; ~
&type plunging anticline; draw noclear
moveitem 'plunging_anticline_approx' to ltype; ~
calc pttype = 'xx'; ~
&type plunging anticline approx; draw noclear
moveitem 'plunging_syncline' to ltype; calc pttype = 'xx'; ~
&type plunging syncline; draw noclear
moveitem 'plunging_syncline_approx' to ltype; ~
calc pttype = 'xx'; ~
&type plunging syncline approx; draw noclear

moveitem 'anticline' to pttype; ~
&type anticline; draw noclear
moveitem 'ot_anticline' to pttype; ~
&type ot_anticline; draw noclear
moveitem 'antiform_axis' to pttype; ~
&type 'antiform_axis'; draw noclear
moveitem 'syncline' to pttype; ~
&type syncline; draw noclear
moveitem 'ot_syncline' to pttype; ~
&type ot_syncline; draw noclear
moveitem 'synform_axis' to pttype; ~
&type 'synform_axis'; draw noclear
moveitem 'fault_ball_fill' to pttype; ~
&type fault_ball_fill; draw noclear
moveitem 'fault_ball_open' to pttype; ~
&type fault_ball_open; draw noclear
moveitem 'fault_ss_rl' to pttype; ~
&type fault_ss_rl; draw noclear
moveitem 'fault_ss_rl_bar' to pttype; ~
&type fault_ss_rl_bar; draw noclear
moveitem 'fault_ss_ll' to pttype; ~
&type fault_ss_ll; draw noclear
moveitem 'fault_ss_ll_bar' to pttype; ~
&type fault_ss_ll_bar; draw noclear
moveitem 'monocline' to pttype; ~
&type monocline; draw noclear
moveitem 'xx' to pttype; ~
&type xx; draw noclear

coordinate mouse
coordinate digitizer
coordinate digitizer %.cover%
digitizer altek /dev/tty00:9600:8bit:none stream
&sv user = [username]; arcplot arc digtest sgm3 ~
/tmp/.dig_signal_%user%:9600:7:even
&sv user = [username]; ~

```

```

'Digi_Server PID'
'New'
'Add'
SEL MENU
'Sel One'
'Select Many'
'Select Box'
'Select Box Pass'
'Sel Path'
'Sel Outline'
'Sel All'
'Aselect Many'
'Unsel'
'Unsel All'
'Nsel'
'Resel le 10'
'Resel le 15'
'Resel le 20'
'Resel le 25'
'Resel le 30'
'Resel le 35'
'Resel le 40'
DEL MENU
'Delete'
'Delete Many'
'Delete One'
'Delete Box'
'Delete Poly'
'Delete Box Pass'
EDIT
'Add'
'Dup Arcs Yes'
'Dup Arcs No'
'Extend One'
'Extend Many'
'Flip'
'Generalize'
'Sel-move'
'Sel-move-node'
'Sel-reshape'
'Sel-snap'
'Sel-split'
'Snap'
'Spline'
'Spline One'
'Spline Many'
'Tiptoe init'
'Tiptoe next'
'Tiptoe clear'
'Unsplit Ltype'
'Vertex Move'
'Vertex Delete'
'Vertex Add'
>Edit Poly'

>Edit Color'
>Edit Ltype'
LIST
'List'
'List One'
'List Many'
'List $all'
PAGEEXT
'Zoom'
'Fullview'
ENV
'AP Off'
'AP Color'
'AP Marker'
digitizer sgmg3 /tmp/.dig_signal_%user%:9600:7:even
&sys ps -ef|grep digi_server|grep -v grep
new
add

sel
sel many
sel box
sel box passthru
sel path
sel outline
sel all
asel many
unsel
unsel all
nsel

resel length le 10
resel length le 15
resel length le 20
resel length le 25
resel length le 30
resel length le 35
resel length le 40

delete
sel many; delete
sel; delete
sel box; delete
sel poly; delete
sel box passthru; delete

add
duplicatearcs yes
duplicatearcs no
sel; extend
sel many; extend
flip; draw
generalize
sel; move
ef node; move; ef arc;
sel; reshape
sel; snap
sel; split
snap
spline
sel; spline
sel many; spline
&r tiptoe init
&r tiptoe next
&r tiptoe clear
unsplit ltype
&r vmov.aml
&r vdel.aml
&r vadd.aml
&return; ef poly; ~
&menu polyedit.menu &stripe polyedit.menu
lineset color; symbolitem arc symbol; draw
lineset geo; symbolitem arc arcrel//symbol; draw

list
sel; list
sel many; list
list $all

pageextent *; draw
pageextent page; draw

ap none; draw
ap none; ap [getfile *.ap.aml 'Select an ap file']; ~
apmode edit; draw
ap none; ap [getfile *.mrk.aml 'Select an ap file']; ~

```

```

apmode edit; draw
&sv .vusav [unq [after [quote [show windows ARCEDIT extent]] :]]
windows extent ARCEDIT %.vusav%
&r dweon.aml; draw
&r dweoff.aml; draw
&r bckcov.aml; draw
&r bckon.aml; draw
&r bckoff.aml; draw
removeback all; draw
&r edfeat.aml
intersectarcs all; de arc intersect; draw
intersectarcs off; de arc tic id; draw
nodecolor dangle blue; nodecolor pseudo green; ~
de nodes errors; draw noclear

DRAW
'Draw'
'Draw Noclear'
'Clear'
'Mape Select'
'Mape *'
'Mape Tic'
OOPS
BLD
SAVE
QT

```

```

1 polyedit.menu
MISC
d          calc ptype = 'd'; calc pattern = 0; ~
&type d; draw noclear
lake       calc ptype = 'lake'; calc pattern = 0; ~
&type lake; draw noclear
mm         calc ptype = 'mm'; calc pattern = 0; ~
&type mm; draw noclear
xx         calc ptype = 'xx'; calc pattern = 9; ~
&type xx; draw noclear
Q-QT
Qa          calc ptype = 'Qa'; calc pattern = 0; ~
&type Qa; draw noclear
Qad         calc ptype = 'Qad'; calc pattern = 101; ~
&type Qad; draw noclear
Qal         calc ptype = 'Qal'; calc pattern = 0; ~
&type Qal; draw noclear
Qas         calc ptype = 'Qas'; calc pattern = 14; ~
&type Qas; draw noclear
Qc          calc ptype = 'Qc'; pattern = 0; ~
&type Qc; draw noclear
Qcr         calc ptype = 'Qcr'; calc pattern = 0; ~
&type Qcr; draw noclear
Qdf         calc ptype = 'Qdf'; calc pattern = 0; ~
&type Qdf; draw noclear
Ql          calc ptype = 'Ql'; calc pattern = 0; ~
&type Ql; draw noclear
Qls         calc ptype = 'Qls'; calc pattern = 0; ~
&type Qls; draw noclear
Qoa         calc ptype = 'Qoa'; calc pattern = 0; ~
&type Qoa; draw noclear
Qs          calc ptype = 'Qs'; calc pattern = 0; ~
&type Qs; draw noclear
Qsm         calc ptype = 'Qsm'; calc pattern = 0; ~
&type Qsm; draw noclear
Qt          calc ptype = 'Qt'; calc pattern = 0; ~
&type Qt; draw noclear
Qtg         calc ptype = 'Qtg'; calc pattern = 0; ~
&type Qtg; draw noclear
Qvr         calc ptype = 'Qvr'; calc pattern = 50; ~
&type Qvr; draw noclear
Qw          calc ptype = 'Qw'; calc pattern = 0; ~
&type Qw; draw noclear
QTa         calc ptype = 'QTa'; calc pattern = 26; ~
&type QTa; draw noclear
QTad        calc ptype = 'QTad'; calc pattern = 0; ~
&type QTad; draw noclear
QTal        calc ptype = 'QTal'; calc pattern = 26; ~
&type QTal; draw noclear
QTam        calc ptype = 'QTam'; calc pattern = 38; ~
&type QTam; draw noclear
QTas        calc ptype = 'QTas'; calc pattern = 0; ~
&type QTas; draw noclear
QTc          calc ptype = 'QTc'; calc pattern = 0; ~
&type QTc; draw noclear
QTcr        calc ptype = 'QTcr'; calc pattern = 38; ~
&type QTcr; draw noclear
QTg          calc ptype = 'QTg'; calc pattern = 0; ~
&type QTg; draw noclear
QTj          calc ptype = 'QTj'; calc pattern = 0; ~
&type QTj; draw noclear
QTjc        calc ptype = 'QTjc'; calc pattern = 102; ~
&type QTjc; draw noclear
QTje        calc ptype = 'QTje'; calc pattern = 101; ~
&type QTje; draw noclear
QTjl        calc ptype = 'QTjl'; calc pattern = 109; ~
&type QTjl; draw noclear
QTp          calc ptype = 'QTp'; calc pattern = 0; ~
&type QTp; draw noclear
QTrg        calc ptype = 'QTrg'; calc pattern = 0; ~
&type QTrg; draw noclear

```

```

QTrl      calc ptype = 'QTrl'; calc pattern = 109; ~
&type QTrl; draw noclear
QTrp      calc ptype = 'QTrp'; calc pattern = 0; ~
&type QTrp; draw noclear
QTs       calc ptype = 'QTs'; calc pattern = 0; ~
&type QTs; draw noclear
Ta-Tdbw
Ta        calc ptype = 'Ta'; calc pattern = 0; ~
&type Ta; draw noclear
Tab        calc ptype = 'Tab'; calc pattern = 0; ~
&type Tab; draw noclear
Tabx      calc ptype = 'Tabx'; calc pattern = 107; ~
&type Tabx; draw noclear
Tad        calc ptype = 'Tad'; calc pattern = 0; ~
&type Tad; draw noclear
Taf        calc ptype = 'Taf'; calc pattern = 0; ~
&type Taf; draw noclear
Tafb      calc ptype = 'Tafb'; calc pattern = 0; ~
&type Tafb; draw noclear
Tam        calc ptype = 'Tam'; calc pattern = 0; ~
&type Tam; draw noclear
Tas        calc ptype = 'Tas'; calc pattern = 50; ~
&type Tas; draw noclear
Tb         calc ptype = 'Tb'; calc pattern = 0; ~
&type Tb; draw noclear
Tbbw      calc ptype = 'Tbbw'; calc pattern = 0; ~
&type Tbbw; draw noclear
Tbci      calc ptype = 'Tbci'; calc pattern = 0; ~
&type Tbci; draw noclear
Tbcv      calc ptype = 'Tbcv'; calc pattern = 0; ~
&type Tbcv; draw noclear
Tbgw      calc ptype = 'Tbgw'; calc pattern = 0; ~
&type Tbgw; draw noclear
Tboa      calc ptype = 'Tboa'; calc pattern = 0; ~
&type Tboa; draw noclear
Tbmp      calc ptype = 'Tbmp'; calc pattern = 0; ~
&type Tbmp; draw noclear
Tbsm      calc ptype = 'Tbsm'; calc pattern = 0; ~
&type Tbsm; draw noclear
Tcbm      calc ptype = 'Tcbm'; calc pattern = 0; ~
&type Tcbm; draw noclear
Tcg       calc ptype = 'Tcg'; calc pattern = 0; ~
&type Tcg; draw noclear
Tcgp      calc ptype = 'Tcgp'; calc pattern = 102; ~
&type Tcgp; draw noclear
Tcr       calc ptype = 'Tcr'; calc pattern = 26; ~
&type Tcr; draw noclear
Tcv       calc ptype = 'Tcv'; calc pattern = 0; ~
&type Tcv; draw noclear
Td        calc ptype = 'Td'; calc pattern = 0; ~
&type Td; draw noclear
Tdbw      calc ptype = 'Tdbw'; calc pattern = 106; ~
&type Tdbw; draw noclear
Tf-Tiw
Tf        calc ptype = 'Tf'; calc pattern = 0; ~
&type Tf; draw noclear
Tfb       calc ptype = 'Tfb'; calc pattern = 0; ~
&type Tfb; draw noclear
Tg        calc ptype = 'Tg'; calc pattern = 0; ~
&type Tg; draw noclear
Tgc       calc ptype = 'Tgc'; calc pattern = 0; ~
&type Tgc; draw noclear
Tgeb      calc ptype = 'Tgeb'; calc pattern = 0; ~
&type Tgeb; draw noclear
Tggw      calc ptype = 'Tggw'; calc pattern = 0; ~
&type Tggw; draw noclear
Tgic      calc ptype = 'Tgic'; calc pattern = 0; ~
&type Tgic; draw noclear
Tgmp      calc ptype = 'Tgmp'; calc pattern = 0; ~
&type Tgmp; draw noclear
Tgoa      calc ptype = 'Tgoa'; calc pattern = 0; ~

```

```

    &type Tgoa; draw noclear
Tgpw    calc ptype = 'Tgpw'; calc pattern = 0; ~
    &type Tgpb; draw noclear
Tgtb    calc ptype = 'Tgpb'; calc pattern = 0; ~
    &type Th; draw noclear
Th      calc ptype = 'Th'; calc pattern = 0; ~
    &type Thb; draw noclear
Thb     calc ptype = 'Thb'; calc pattern = 0; ~
    &type Thci; draw noclear
Thci    calc ptype = 'Thci'; calc pattern = 103; ~
    &type Thcv; draw noclear
Thcv    calc ptype = 'Thcv'; calc pattern = 0; ~
    &type Thl; draw noclear
Thl     calc ptype = 'Thl'; calc pattern = 0; ~
    &type Thll; draw noclear
Thll    calc ptype = 'Thll'; calc pattern = 0; ~
    &type Thlu; draw noclear
Thlu    calc ptype = 'Thlu'; calc pattern = 0; ~
    &type Thlv; draw noclear
Thlv    calc ptype = 'Thlv'; calc pattern = 0; ~
    &type Thlv; draw noclear
    calc ptype = 'Thr'; calc pattern = 0; ~
    &type Thrg; draw noclear
Thrg    calc ptype = 'Thrg'; calc pattern = 101; ~
    &type Thtg; draw noclear
Thtg    calc ptype = 'Thtg'; calc pattern = 101; ~
    &type Thtc; draw noclear
Thtc    calc ptype = 'Thtc'; calc pattern = 102; ~
    &type Thtb; draw noclear
Thtb    calc ptype = 'Thtb'; calc pattern = 0; ~
    &type Thht; draw noclear
Thht    calc ptype = 'Thht'; calc pattern = 0; ~
    &type Thtg; draw noclear
Thtg    calc ptype = 'Thtg'; calc pattern = 101; ~
    &type Ths; draw noclear
Ths     calc ptype = 'Ths'; calc pattern = 0; ~
    &type Ti; draw noclear
Ti      calc ptype = 'Ti'; calc pattern = 0; ~
    &type Tid; draw noclear
Tid     calc ptype = 'Tid'; calc pattern = 0; ~
    &type Tiw; draw noclear
Tiw     calc ptype = 'Tiw'; calc pattern = 0; ~
    &type Tiw; draw noclear

Tm-Trgw
Tm      calc ptype = 'Tm'; calc pattern = 0; ~
    &type Tm; draw noclear
Tmc1    calc ptype = 'Tmc1'; calc pattern = 0; ~
    &type Tmc1; draw noclear
Tmcu    calc ptype = 'Tmcu'; calc pattern = 0; ~
    &type Tmcu; draw noclear
Tmc     calc ptype = 'Tmc'; calc pattern = 102; ~
    &type Tmc; draw noclear
Tmi     calc ptype = 'Tmi'; calc pattern = 0; ~
    &type Tmi; draw noclear
Tmg     calc ptype = 'Tmg'; calc pattern = 0; ~
    &type Tmg; draw noclear
Tmgl    calc ptype = 'Tmgl'; calc pattern = 0; ~
    &type Tmgl; draw noclear
Tmgm    calc ptype = 'Tmgm'; calc pattern = 0; ~
    &type Tmgm; draw noclear
Tml     calc ptype = 'Tml'; calc pattern = 0; ~
    &type Tml; draw noclear
Tmms    calc ptype = 'Tmms'; calc pattern = 0; ~
    &type Tmms; draw noclear
Tmss    calc ptype = 'Tmss'; calc pattern = 0; ~
    &type Tmss; draw noclear
Tmsu    calc ptype = 'Tmsu'; calc pattern = 0; ~
    &type Tmsu; draw noclear
Tmt     calc ptype = 'Tmt'; calc pattern = 0; ~
    &type Tmt; draw noclear
Tmu     calc ptype = 'Tmu'; calc pattern = 0; ~
    &type Tmu; draw noclear
Tob     calc ptype = 'Tob'; calc pattern = 0; ~

```

```

Tpm          &type Tob; draw noclear
Tpm          calc ptype = 'Tpm'; calc pattern = 0; ~
Tpm1         &type Tpm; draw noclear
Tpm1         calc ptype = 'Tpm1'; calc pattern = 0; ~
Tpm2         &type Tpm1; draw noclear
Tpm2         calc ptype = 'Tpm2'; calc pattern = 0; ~
Tpm3         &type Tpm2; draw noclear
Tpm3         calc ptype = 'Tpm3'; calc pattern = 0; ~
Tpmbx        &type Tpm3; draw noclear
Tpmbx        calc ptype = 'Tpmbx'; calc pattern = 94; ~
Tpms         &type Tpmbx; draw noclear
Tpms         calc ptype = 'Tpms'; calc pattern = 101; ~
Tpp          &type Tpms; draw noclear
Tpp          calc ptype = 'Tpp'; calc pattern = 0; ~
Tr           &type Tpp; draw noclear
Tr           calc ptype = 'Tr'; calc pattern = 0; ~
&type Tr; draw noclear
Trb          calc ptype = 'Trb'; calc pattern = 0; ~
&type Trb; draw noclear
Trbi         calc ptype = 'Trbi'; calc pattern = 0; ~
&type Trbi; draw noclear
Trc          calc ptype = 'Trc'; calc pattern = 102; ~
&type Trc; draw noclear
Trgw         calc ptype = 'Trgw'; calc pattern = 0; ~
&type Trgw; draw noclear
Trgwc-Twr
Trgwc        calc ptype = 'Trgwc'; calc pattern = 102; ~
&type Trgwc; draw noclear
Tri          calc ptype = 'Tri'; calc pattern = 0; ~
&type Tri; draw noclear
Trmb         calc ptype = 'Trmb'; calc pattern = 0; ~
&type Trmb; draw noclear
Trmd         calc ptype = 'Trmd'; calc pattern = 51; ~
&type Trmd; draw noclear
Trmi         calc ptype = 'Trmi'; calc pattern = 103; ~
&type Trmi; draw noclear
Trmi2        calc ptype = 'Trmi2'; calc pattern = 104; ~
&type Trmi2; draw noclear
Trmv         calc ptype = 'Trmv'; calc pattern = 39; ~
&type Trmv; draw noclear
Trmzb        calc ptype = 'Trmzb'; calc pattern = 27; ~
&type Trmzb; draw noclear
Trs          calc ptype = 'Trs'; calc pattern = 0; ~
&type Trs; draw noclear
Trsc         calc ptype = 'Trsc'; calc pattern = 0; ~
&type Trsc; draw noclear
Trsb         calc ptype = 'Trsb'; calc pattern = 0; ~
&type Trsb; draw noclear
Trsbl        calc ptype = 'Trsbl'; calc pattern = 0; ~
&type Trsbl; draw noclear
Trsbu        calc ptype = 'Trsbu'; calc pattern = 0; ~
&type Trsbu; draw noclear
Trsc         calc ptype = 'Trsc'; calc pattern = 0; ~
&type Trsc; draw noclear
Ts           calc ptype = 'Ts'; calc pattern = 0; ~
&type Ts; draw noclear
Tsgp         calc ptype = 'Tsgp'; calc pattern = 0; ~
&type Tsgp; draw noclear
Tsl          calc ptype = 'Tsl'; calc pattern = 95; ~
&type Tsl; draw noclear
Tss          calc ptype = 'Tss'; calc pattern = 0; ~
&type Tss; draw noclear
Ttbs         calc ptype = 'Ttbs'; calc pattern = 0; ~
&type Ttbs; draw noclear
Tthd         calc ptype = 'Tthd'; calc pattern = 0; ~
&type Tthd; draw noclear
Ttl          calc ptype = 'Ttl'; calc pattern = 0; ~
&type Ttl; draw noclear
Tvc          calc ptype = 'Tvc'; calc pattern = 0; ~
&type Tvc; draw noclear
Tvcl         calc ptype = 'Tvcl'; calc pattern = 0; ~

```

```

Tvc2          &type Tvc1; draw noclear
              calc ptype = 'Tvc2'; calc pattern = 0; ~
Tvc3          &type Tvc2; draw noclear
              calc ptype = 'Tvc3'; calc pattern = 0; ~
Tvc4          &type Tvc3; draw noclear
              calc ptype = 'Tvc4'; calc pattern = 0; ~
Tvh           &type Tvc4; draw noclear
              calc ptype = 'Tvh'; calc pattern = 0; ~
Tvhl          &type Tvh; draw noclear
              calc ptype = 'Tvhl'; calc pattern = 0; ~
Tvhm          &type Tvhl; draw noclear
              calc ptype = 'Tvhm'; calc pattern = 0; ~
Tvhu          &type Tvhm; draw noclear
              calc ptype = 'Tvhu'; calc pattern = 0; ~
Twr           &type Tvhu; draw noclear
              calc ptype = 'Twr'; calc pattern = 0; ~
&type Twr; draw noclear

MESOZOIC
Kb            calc ptype = 'Kb'; calc pattern = 0; ~
&type Kb; draw noclear
Kbc           calc ptype = 'Kbc'; calc pattern = 102; ~
&type Kbc; draw noclear
Kbs           calc ptype = 'Kbs'; calc pattern = 0; ~
&type Kbs; draw noclear
Kw            calc ptype = 'Kw'; calc pattern = 0; ~
&type Kw; draw noclear
Kwt           calc ptype = 'Kwt'; calc pattern = 0; ~
&type Kwt; draw noclear
Ja             calc ptype = 'Ja'; calc pattern = 0; ~
&type Ja; draw noclear
Jmk           calc ptype = 'Jmk'; calc pattern = 0; ~
&type Jmk; draw noclear
JT Ra          calc ptype = 'JT Ra'; calc pattern = 0; ~
&type JT Ra; draw noclear
JT Rmk         calc ptype = 'JT Rmk'; calc pattern = 0; ~
&type JT Rmk; draw noclear
TRc            calc ptype = 'TRc'; calc pattern = 0; ~
&type TRc; draw noclear
TRm           calc ptype = 'TRm'; calc pattern = 0; ~
&type TRm; draw noclear

PALEOZOIC
PPb           calc ptype = 'PPb'; calc pattern = 0; ~
&type PPb; draw noclear
PPc           calc ptype = 'PPc'; calc pattern = 0; ~
&type PPc; draw noclear
PPpc          calc ptype = 'PPpc'; calc pattern = 0; ~
&type PPpc; draw noclear
Pc             calc ptype = 'Pc'; calc pattern = 0; ~
&type Pc; draw noclear
Pe             calc ptype = 'Pe'; calc pattern = 0; ~
&type Pe; draw noclear
Pep            calc ptype = 'Pep'; calc pattern = 0; ~
&type Pep; draw noclear
Ph             calc ptype = 'Ph'; calc pattern = 0; ~
&type Ph; draw noclear
Phe            calc ptype = 'Phe'; calc pattern = 0; ~
&type Phe; draw noclear
Pk             calc ptype = 'Pk'; calc pattern = 0; ~
&type Pk; draw noclear
Pkt            calc ptype = 'Pkt'; calc pattern = 0; ~
&type Pkt; draw noclear
Pr             calc ptype = 'Pr'; calc pattern = 0; ~
&type Pr; draw noclear
Prb            calc ptype = 'Prb'; calc pattern = 0; ~
&type Prb; draw noclear
MDs            calc ptype = 'MDs'; calc pattern = 0; ~
&type MDs; draw noclear
Mps            calc ptype = 'Mps'; calc pattern = 0; ~
&type Mps; draw noclear
Mm             calc ptype = 'Mm'; calc pattern = 0; ~
&type Mm; draw noclear

```

```

Mr          calc ptype = 'Mr'; calc pattern = 0; ~
&type Mr; draw noclear
Ms          calc ptype = 'Ms'; calc pattern = 0; ~
&type Ms; draw noclear
Dtb         calc ptype = 'Dtb'; calc pattern = 0; ~
&type Dtb; draw noclear
Om          calc ptype = 'Om'; calc pattern = 0; ~
&type Om; draw noclear
C           calc ptype = 'C'; calc pattern = 0; ~
&type C; draw noclear
Ca          calc ptype = 'Ca'; calc pattern = 0; ~
&type Ca; draw noclear
Cb          calc ptype = 'Cb'; calc pattern = 0; ~
&type Cb; draw noclear
Cba         calc ptype = 'Cba'; calc pattern = 0; ~
&type Cba; draw noclear
Cbk         calc ptype = 'Cbk'; calc pattern = 0; ~
&type Cbk; draw noclear
Cd          calc ptype = 'Cd'; calc pattern = 0; ~
&type Cd; draw noclear
Cm          calc ptype = 'Cm'; calc pattern = 0; ~
&type Cm; draw noclear
Ct          calc ptype = 'Ct'; calc pattern = 0; ~
&type Ct; draw noclear
Cu          calc ptype = 'Cu'; calc pattern = 0; ~
&type Cu; draw noclear

PROTEROZOIC
pC          calc ptype = 'pC'; calc pattern = 0; ~
&type pC; draw noclear
Yd          calc ptype = 'Yd'; calc pattern = 0; ~
&type Yd; draw noclear
Yg          calc ptype = 'Yg'; calc pattern = 0; ~
&type Yg; draw noclear
Ygp         calc ptype = 'Ygp'; calc pattern = 105; ~
&type Ygp; draw noclear
Yge         calc ptype = 'Yge'; calc pattern = 103; ~
&type Yge; draw noclear
Ygm         calc ptype = 'Ygm'; calc pattern = 104; ~
&type Ygm; draw noclear
YXb         calc ptype = 'YXb'; calc pattern = 0; ~
&type YXb; draw noclear
YXd         calc ptype = 'YXd'; calc pattern = 0; ~
&type YXd; draw noclear
YXdg        calc ptype = 'YXdg'; calc pattern = 0; ~
&type YXdg; draw noclear
Xa          calc ptype = 'Xa'; calc pattern = 0; ~
&type Xa; draw noclear
Xd          calc ptype = 'Xd'; calc pattern = 0; ~
&type Xd; draw noclear
Xg          calc ptype = 'Xg'; calc pattern = 0; ~
&type Xg; draw noclear
Xgc         calc ptype = 'Xgc'; calc pattern = 0; ~
&type Xgc; draw noclear
Xgg         calc ptype = 'Xgg'; calc pattern = 0; ~
&type Xgg; draw noclear
Xgp         calc ptype = 'Xgp'; calc pattern = 0; ~
&type Xgp; draw noclear
Xgq         calc ptype = 'Xgq'; calc pattern = 103; ~
&type Xgq; draw noclear
Xgr         calc ptype = 'Xgr'; calc pattern = 0; ~
&type Xgr; draw noclear
Xhl         calc ptype = 'Xhl'; calc pattern = 103; ~
&type Xhl; draw noclear
Xlg         calc ptype = 'Xlg'; calc pattern = 0; ~
&type Xlg; draw noclear
Xm          calc ptype = 'Xm'; calc pattern = 0; ~
&type Xm; draw noclear
Xqd         calc ptype = 'Xqd'; calc pattern = 104; ~
&type Xqd; draw noclear
Xqs         calc ptype = 'Xqs'; calc pattern = 105; ~
&type Xqs; draw noclear

```

```

Xu           calc ptype = 'Xu'; calc pattern = 0; ~
&type Xu; draw noclear
Xwe          calc ptype = 'Xwe'; calc pattern = 0; ~
&type Xwe; draw noclear
Xwp          calc ptype = 'Xwp'; calc pattern = 0; ~
&type Xwp; draw noclear

'SEL MENU'
'Sel One'      sel
>Select Many'   sel many
>Select All'    sel all
'Asel Many'    asel many
'Unsel'        unsel
'Unsel All'    unsel all

LIST
>List'          list
>List Many'     sel many; list
>List One'      sel; list

ZOOM-FLVW
'Zoom'          pageextent *; draw
'Flvw'          pageextent page; draw

DRAW
'Draw'          draw
'Draw Noclear' draw noclear

ENV
'Merge'         merge
'AP Color'      ap [getfile *.ap.aml 'Select an ap file']; ~
apmode edit; draw
'AP Pattern'    ap [getfile *.pat.aml 'Select pattern ap file']; ~
apmode edit; draw
'AP Off'        ap none; draw
'Poly Fill'     de poly fill; draw
'Tiptoe init'   &r tiptoe init
'Tiptoe next'   &r tiptoe next
'Tiptoe clear'  &r tiptoe clear
&return; ef arcs; ~

EDITARCS
&menu /arcwork/menus/arcedit.menu &stripe arcedit.menu

COMMAND
&sv command [response 'Type Command']; ~
&lv command; ~
[unq %command%]
oops

OOPS
SAVE          save; ec %.cover%; ef poly; &type all saved!
QT            &type GoodBye!!; &return; quit

```

ATTACHMENT 9

```
/*
 * Command name: TIPTOE (Version 2.3)
 * Language: AML for ARCEDIT
 * Pathname on UNIX computer: $ARCHOME/atool/arcedit/tiptoe.aml
 *
 ** ****
 */
/* DISCLAIMER:
 * THIS VERSION DATED 04/28/92,
 * WRITTEN BY KENN D. CARTIER, U.S. GEOLOGICAL SURVEY.
 * THE AUTHOR HAS TESTED THIS PROGRAM FOR ERRORS. HOWEVER, THE U.S.
 * GEOLOGICAL SURVEY MAKES NO GUARANTEE OR WARRANTY, EXPRESSED OR,
 * IMPLIED REGARDING THE RELIABILITY OF THIS PROGRAM, AND ASSUMES NO
 * RESPONSIBILITY FOR ITS USE OR MODIFICATION.
 ** ****
 */
/*::::::::::::
 */
/* Purpose:
 * TIPTOE sequentially establishes MAPEXTENTS and reDRAWS the coverage
 * around each element in the set of currently selected elements.
 * TIPTOE uses the set of elements that were SELECTed at the time
 * TIPTOE was initially invoked. TIPTOE steps through the set of
 * selected elements, one element at a time.
 */
/* Macro is useful for systematically examining each element
 * in the selected set. It is most useful for examining
 * a few elements (less than 30), such as short arcs, which would
 * otherwise be difficult to locate.
 */
/*::::::::::::
 */
/* Usage:
 * TIPTOE <INITIAL | NEXT | REINIT | CLEAR> {mape_side_length} {OVIEW}
 */
/*::::::::::::
 */
/* Arguments
 * Name,      Type,    Definition
 * -----
 * <function>  C      TIPTOE functions.
 *               Choices: INITIAL | NEXT | REINIT | CLEAR.
 *               INITIAL - Establishes TIPTOE environment and
 *                         stores the record number for the initially
 *                         selected elements in a set of global
 *                         variables.
 *               NEXT - Advances mapextent to next element of
 *                         the selected set.
 *               REINIT - Reinitializes sequence to element
 *                         with the lowest record number in the
 *                         selected set.
 *               CLEAR - Clears all global and pseudo
 *                         variables used by the TIPTOE macro.
 * {mape_side}  N      The length of the sides of the mapextent window,
 *                     in coverage units, used for DRAWing the cover.
 *                     The # sign can be used to indicate that default
 *                     value will be used.
 *                     For EDITFEATURE equal to ARC:
 *                     DEFAULT = mapextent large enough to completely
 *                     enclose selected arc.
 *                     For EDITFEATURE not equal to ARC:
 *                     DEFAULT = 5% of average of coverage width and
 *                     height.
 * {icon_draw}   C      Indicates whether the OVVIEW overview icon should
 *                     be drawn. Default is to not draw the icon.
 */
/* History:
 * Author/Site,          Date,           Event
 */
```

```

/*
-----  

/*      KDCartier, USGS-NV WRD    12/20/90    Initial AML coding. (1.0)  

/*                                04/20/91    Modified.          (2.0)  

/*                                07/10/91    Changed error handling (2.1)  

/*                                04/28/92    Modified to UNIX. (2.2)  

/*                                09/20/92    Added element# display (2.3)  

/*:::::::::::::::::::::::::::::::::::  

/*  

/* Notes:  

/*      o The number of selected elements must be less than 100.  

/*  

/*      o Macro draws EDITCOVER features with the following linecolors:  

/*          1) Red - currently selected element  

/*          2) Green - elements which were in the selected set  

/*          3) White - nonselected elements  

/*  

/*      o Macro calls OVVIEW AML for DRAWING the coverage, if the OVVIEW  

/* macro is on system. OVVIEW and OVVIEW_AP must be located in the  

/* ATOOL directory for ARCEDIT AMLs. The OVVIEW AMLs will both  

/* draw coverage and an OVERVIEW icon. The icon is very useful for  

/* identifying the current mapextent location relative to the  

/* default mapextent.  

/*-----  

/*  

/* Disclaimer:  

/*      Although this program has been used by the U.S. Geological Survey, no  

/* warranty, expressed or implied, is made by the USGS as to the accuracy  

/* and functioning of the program and related program material nor shall the  

/* fact of distribution constitute any such warranty, and no responsibility  

/* is assumed by the USGS in connection herewith.  

/*-----  

/*-----  

/*-----  

/*      -- ENTER AND CONFIRM ARGUMENTS  

&ARG function mape_side icon_draw  

&CALL sub_confirmargs  

/*  

/*      -- DEFINE SYSTEM SPECIFIC VARIABLE -----  

/*&S atoolpath = $ARCHOME/atool/arcedit /* Pathname to ATOOL directory  

&sv atoolpath = /arcwork/atool/arcedit /* Modified for hydra  

/*-----  

/*-----  

/*      -- ESTABLISH ERROR HANDLING  

&SEVERITY &ERROR &ROUTINE sub_error  

/*  

/*      -- CHECK THAT TIPTOE ENVIRONMENT HAS BEEN INITIALIZED  

&IF %key_function% > 1 &THEN &DO /* For keywords other than "INIT"  

  &IF NOT [VARIABLE .tot_selected] &THEN &DO  

    &RETURN &WARNING \Please initialize TIPTOE by invoking 'TIPTOE INIT'  

  &END  

&END  

/*-----  

/*      -- PERFORM THE SPECIFIED FUNCTION  

&SELECT %key_function%  

&WHEN 1 /* FOR FUNCTION "INIT"  

  &DO  

  /*  

  /*      -- TYPE TIPTOE HEADER  

  &TYPE \TIPTOE (REV 2.3) KDC  

  /*  

  /*      Delete TIPTOE variables which may be remaining from previous executions  

  &IF [VARIABLE .tot_selected] &THEN &DELVAR .tt_element*  

  /*  

  /*      Determine number of currently selected elements and save to a global  

  /*      variable for future reference by macro  

  &S .tot_selected = [SHOW NUMBER SELECT]  

  &TYPE \Number of elements in initial selected set: %.tot_selected%  

  /*  

  /*      Confirm that at least one element is now selected  

  &IF %.tot_selected% > 0 &THEN &DO  

  /*

```

```

/* Check that number of selected elements is not greater than 99
&IF %.tot_selected% > 99 &THEN &DO
  &RETURN &WARNING Number of selected elements must be less than 100.
&END
/*
&TYPE \Please wait...
/*
/* Store record numbers for the initial selected set to a set
/* of global variables for future reference by macro
&DO I = 1 &TO %.tot_selected%
  &S .tt_element%i% = [SHOW SELECT %i%]
&END
/*
/* Set color variable for selected set to 3.
GRAPHICS OFF
CALC $SYMBOL = 3
&MESSAGES &OFF
NSELECT
&MESSAGES &ON
/*
/* Set color variable for non-select set to 1
CALC $SYMBOL = 1
&MESSAGES &OFF
NSELECT
&MESSAGES &ON
GRAPHICS ON
/*
/* Initialize the record-number pointer for the initial selected set
&S .curr_select = 0
&END
&ELSE &DO /* If no elements are selected, then exit.
  &RETURN &WARNING No elements are selected ..
&END
&END
&WHEN 3 /* FUNCTION "REINIT"
&DO
  &TYPE \Please wait...
/*
/* Save the number of initially selected elements to a temporary variable
&S old_select = %.tot_selected%
/*
/* Reselect the elements which were in the initial select set.
/* NOTE: Arcs which have been deleted will be ignored.
&MESSAGES &OFF
SELECT $SYMBOL > 1
&MESSAGES &ON
/*
/* Determine number of currently selected elements and save to a global
/* variable for future reference by macro
&S .tot_selected = [SHOW NUMBER SELECT]
&TYPE \Number of elements in new selected set: %.tot_selected%
/*
&IF %.tot_selected% > 0 &THEN &DO /* Confirm that at least one element is
  /* now selected
    /* Set color variable for new selected set to 3.
    GRAPHICS OFF
    CALC $SYMBOL = 3
    GRAPHICS ON
    /*
    /* Store record numbers for the new initial select set to a set
    /* of global variables for future reference by macro
    &DO I = 1 &TO %.tot_selected%
      &S .tt_element%i% = [SHOW SELECT %i%]
    &END
    /*
    /* Delete unneeded variables
    &IF %.tot_selected% < %old_select% &THEN &DO
      &S start = [CALC %.tot_selected% + 1 ]
      &DO I = %start% &TO %old_select%
        &DELVAR .tt_element%i%
    &END

```

```

&END
/*
/* Re-initialize the record-number pointer for the new selected set
&S .curr_select = 0
&END
&ELSE &DO /* If no elements are selected, then exit.
  &RETURN &WARNING No elements could be re-initialized..
&END
&END
&WHEN 4 /* FOR FUNCTION "CLEAR"
&DO
  &TYPE Deleting all variables set by TIPTOE
  &TYPE \Please wait...
/*
/* Set color variable for all elements to 1
&MESSAGES &OFF
SELECT $SYMBOL = 2 OR $SYMBOL = 3
&MESSAGES &ON
CALC $SYMBOL = 1
/*
/* Delete variables used by TIPTOE
&DELVAR .tot_selected .curr_select .tt_element*
/*
/* Exit from macro
&RETURN &WARNING Done...
&END
&END
/*
/* -- ADVANCE THE RECORD-NUMBER POINTER FOR THE SELECTED SET AND CONFIRM
/* -- THAT THERE ARE MORE ELEMENTS TO BE DRAWN
&S .curr_select = [CALC %.curr_select% + 1 ]
&IF %.curr_select% > %.tot_selected% &THEN &DO
  &RETURN &WARNING No more selected elements.
&END
/*
/* -- SELECT THE NEXT ELEMENT IN THE SELECTED SET
&TYPE Selecting element %.curr_select% of %.tot_selected%.
SELECT $RECNO = [VALUE .tt_element%.curr_select% ]
/*
/* -- SET THE MAPEXTENT TO BE CENTERED ON THE CURRENTLY SELECTED ELEMENT
/* -- OF THE SELECTED SET
MAPEXTENT SELECT
/*
/* -- FOR DEFAULT MAPEXTENT AND EDITFEATURE NOT EQUAL TO ARC,
/* -- SET MAPEXTENT EQUAL TO 5% OF AVERAGE WIDTH AND HEIGHT OF COVERAGE
&IF [QUOTE %mape_side%] = 'DEFAULT' AND [SHOW EDITFEAT] NE ARC &THEN
&DO
  &S percent = 5 /* Set percentage value to use
  &S half_percent = [CALC %percent% / 2 ]
  &DESCRIBE [ SHOW EDITCOVERAGE ]
  &S lenx = [CALC %DSC$XMAX% - %DSC$XMIN% ]
  &S leny = [CALC %DSC$YMAX% - %DSC$YMIN% ]
  &S mape_dim = [CALC ( ( %lenx% + %leny% ) / 200 ) * %half_percent% ]
  &S mape_dim = [CALC [ROUND [CALC %mape_dim% * 10000 ] ] / 10000 ]
&END
/*
/* -- FOR USER SPECIFIED MAPEXTENT VALUE OR DEFAULT MAPEXTENT FOR EDITFEATURE
/* -- NOT EQUAL TO ARC, SET MAPEXTENT USING MAPE_DIM VALUE SPECIFIED BY USER
&IF NOT ( [QUOTE %mape_side%] = 'DEFAULT' AND [SHOW EDITFEAT] = ARC ) &THEN
&DO
  &S sel_mape = [SHOW MAPE]
  &S x_cen = [CALC [CALC [EXTRACT 1 %sel_mape%] + [EXTRACT 3 %sel_mape%]] / 2 ]
  &S y_cen = [CALC [CALC [EXTRACT 2 %sel_mape%] + [EXTRACT 4 %sel_mape%]] / 2 ]
  &S xmin = [CALC %x_cen% - %mape_dim%]
  &S xmax = [CALC %x_cen% + %mape_dim%]
  &S ymin = [CALC %y_cen% - %mape_dim%]
  &S ymax = [CALC %y_cen% + %mape_dim%]
  MAPE %xmin% %ymin% %xmax% %ymax%
&END
/*
/* -- DETERMINE MAPEXTENT DIMENSIONS

```

```

&S curr_mape = [SHOW MAPEXTENT]
&S x_dim = [CALC [EXTRACT 3 %curr_mape%] - [EXTRACT 1 %curr_mape%]]
&S y_dim = [CALC [EXTRACT 4 %curr_mape%] - [EXTRACT 2 %curr_mape%]]
/*
/* -- DRAW THE COVERAGE FOR THE SPECIFIED MAPEXTENT
/* -- MACRO USES THE OVVIEW AML, IF IT EXISTS IN THE ATOOL DIRECTORY
/* -- AND THE USER HAS SELECTED THIS OPTION
&IF [EXISTS %atoolpath%/OVVIEW.AML] AND %key_icon_draw% = 2 &THEN
    OVVIEW
&ELSE DRAW
/*
/* -- REDRAW THE CURRENTLY SELECTED ELEMENT IN COLOR 2
SETDRAW 2
DRAWSELECT
/*
/* -- DISPLAY DESCRIPTION OF COLOR FORMAT
&TYPE \TIPTOE Colors:
&TYPE     RED = currently selected element, GREEN = other elements in selected set
&TYPE     WHITE = nonselected elements
&TYPE Mapextent dimensions: %x_dim% by %y_dim%
/*
/* -- EXIT AND RETURN TO ARCEDIT PROMPT
&RETURN
/*
/*
/* --- SUB-ROUTINES -----
&ROUTINE sub_error
    /* HANDLE BAILOUT ERRORS
    &MESSAGES &ON
    &TYPE Error encountered. Exiting...
&RETURN;&RETURN;&RETURN;&RETURN
/*
&ROUTINE sub_confirmargs
    /* CONFIRM ARGUMENT ENTRY
    /*
    /* Check FUNCTION argument
    &IF [NULL %function%] &THEN &CALL sub_usage
    &S key_function = [KEYWORD %function% INITIAL NEXT REINIT CLEAR]
    &IF %key_function% < 1 &THEN &CALL sub_usage
    /*
    /* Check MAPE_SIDE argument
    &IF [NULL %mape_side%] &THEN &S mape_side = #
    &IF [QUOTE %mape_side%] = '#' &THEN &S mape_side = DEFAULT
    &IF [TYPE %mape_side%] = 1 &THEN &DO /* For DEFAULT
        &IF [KEYWORD %mape_side% DEFAULT] = 1 &THEN &S mape_side = DEFAULT
        &ELSE &CALL sub_usage
    &END
    &ELSE &DO /* For a specified mapextent size
        &IF [TYPE %mape_side%] = 2 &THEN &CALL sub_usage
        &S mape_dim = [CALC %mape_side% / 2]
    &END
    /*
    /* Check ICON_DRAW argument
    &IF [NULL %icon_draw%] &THEN &S icon_draw = #
    &IF [QUOTE %icon_draw%] = '#' &THEN &S icon_draw = NOOVVIEW
    &S key_icon_draw = [KEYWORD %icon_draw% NOOVVIEW OVVIEW]
    &IF %key_icon_draw% < 1 &THEN &CALL sub_usage
&RETURN
/*
&ROUTINE sub_usage
    /* DISPLAY USAGE
    /*
    &TYPE Usage TIPTOE <INITIAL | NEXT | REINIT | CLEAR> {mape_side_length}~
    {OVVIEW}
&RETURN;&RETURN;&RETURN
/*
/* -- EOP

```

ATTACHMENT 10

```
/* ADDITEMFOLD.AML
/*
/* Creates fold coverage from selected tic_bnd_cover,
/* adds items to .aat file and extracts data.
/*
/* Checks for lookup tables and relate file.

&if [exists geomrk.lut -info] AND ~
  [exists geolin.lut -info] AND ~
  [exists geo.rel -info] &then

&do
  &sv bndcov = [getcover * -all 'Select tic_bnd_cover']
  &sv cover = [response 'Enter name of new fold cover']
  create %cover% %bndcov%
  ae
  MAPE %cover%
  EC %cover%

  EF arc
  CREATEATTRIBUTES
  ADDITEM ltype 35 35 c
  ADDITEM pttype 35 35 c
  ADDITEM plunge 3 3 i
  save
  quit
&end

build %cover% line

&do
  &sv pcover = [getcover * -poly 'Select name of poly cover with fold data']
  EC %pcover%
  DE arc
  DRAW
  EF arc
  SEL LTYPE cn 'dike'
  &if [show number select] > 0 &then &do
    put %cover%
    save
  &end
  SEL LTYPE cn 'anticline'
  &if [show number select] > 0 &then &do
    put %cover%
    save
  &end
  SEL LTYPE cn 'syncline'
  &if [show number select] > 0 &then &do
    put %cover%
    save
  &end
  quit
&end

&else
  &type Lookup tables and relate file must be in INFO directory! RUN NEWLUT.AML!

&return
```

ATTACHMENT 11

```
/* ADDITEMDIP.AML
/*
/* Creates dip coverage from selected tic_bnd_cover
/* adds items to .pat file.
/*
/* Checks for lookup table and relate file.

&if [exists geomrk.lut -info] AND ~
    [exists geo.rel -info] &then

&do
    &sv bndcov = [getcover * -all 'Select tic_bnd cover']
    &sv cover = [response 'Enter name of new dip cover']
    create %cover% %bndcov%
    AE
    MAPE %cover%
    EC %cover%

    EF point
    CREATEATTRIBUTES
    ADDITEM pttype 35 35 c
    ADDITEM dip 3 3 i
    ADDITEM strike 3 3 i

    EF arc
    CREATEATTRIBUTES

    save
    quit
&end

&else
    &type Lookup tables and relate file must be in INFO directory! RUN NEWLUT.AML!

&return
```

ATTACHMENT 12

```
/* ATTITUDE.AML
/* AML to invoke menu for adding bedding, fault dip, foliation symbols

display 9999 size 1280 650 pos 100 75
ae
&sv cover = [GETCOVER * -point 'Select a dip coverage']

mape %cover%
mapunits meters
&sv .mapscale = [RESPONSE 'Enter the mapscale']
mapscale %.mapscale%
ec %cover%

lineset geo.lin
textset font
markerset geo.mrk

relate restore geo.rel
symbolitem point dipptrel//symbol
de point anno
ef point

&sv .termflag = 99           /* Allows for loop when adding markers.
&sv .plungeflag = 98

&sv bcov1 = [GETCOVER * -poly 'Select a poly backcoverage' -none]
&if NOT [NULL %bcov1%] &then
  &do
    bc %bcov1%
    backsymbolitem %bcov1% arc arcrel//symbol
    be %bcov1% arc
    ap [GETFILE *.mrk.aml 'Select file to display arcmarkers' -none]
  &end
&sv bcov2 = [GETCOVER * -arc 'Select a fold backcoverage' -none]
&if NOT [NULL %bcov2%] &then
  &do
    bc %bcov2%
    backsymbolitem %bcov2% arc arcrel//symbol
    be %bcov2% arc
  &end
&sv bcov3 = [GETCOVER * -anno 'Select an anno backcoverage' -none]
&if NOT [NULL %bcov3%] &then
  &do
    bc %bcov3%
    backsymbolitem %bcov3% arc arcrel//symbol
    be %bcov3% arc anno.unit
  &end

editdistance default
setdrawsymbol 0 magenta
drawsel all
draw

&terminal 9999
&fullscreen &popup
&menu /arcwork/menus/attitude.menu &stripe attitude.menu

&return
```

```

1 attitude.menu

COMMAND          &sv command [response 'Type a command']; ~
                  &lv command; ~
                  [unq %command%]

SNAPPING
'Set snapping'   &sv snaptol = %.mapscale% * .0254 * .05
'Snap to BC'      sc %backcover%; sf point arc; snapping closest %snaptol%
'Snap to EC'      sc %cover%; sf point point; snapping closest %snaptol%
'Sel snap'        sel; snap
'Snap'            snap

BEDDING
'Bedding'         new; moveitem 'bedding' to pttype; &r dip.aml
'Vert Bed'        new; moveitem 'vertical_bedding' to pttype; &r dip.aml
'Horz Bed'        new; moveitem 'horizontal_bedding' to pttype; add
'Approx Bed'      new; moveitem 'approx_bedding' to pttype; &r dip.aml
'OVERTURNED'      new; moveitem 'ot_bedding' to pttype; &r dip.aml

FOLIATION
'Foliation'       new; moveitem 'foliation' to pttype; &r dip.aml
'Fol and Bed'    new; moveitem 'foliation_and_bedding' to pttype; &r dip.aml
'Vert Fol'        new; moveitem 'vertical_foliation' to pttype; &r dip.aml
'Horz Fol'        new; moveitem 'horizontal_foliation' to pttype; &r dip.aml
'Ign Fol'         new; moveitem 'igneous_foliation' to pttype; &r dip.aml
'Ign Vert Fol'   new; moveitem 'igneous_vertical_foliation' to pttype; &r dip.aml
'Ign Horz Fol'   new; moveitem 'igneous_horizontal_foliation' to pttype; &r dip.aml
'Mag Fol'         new; moveitem 'magmatic_foliation' to pttype; &r dip.aml
'Mag Vert Fol'   new; moveitem 'magmatic_vertical_foliation' to pttype; &r dip.aml
'Mag Horz Fol'   new; moveitem 'magmatic_horizontal_foliation' to pttype; &r dip.aml
'Meta Fol'        new; moveitem 'metamorphic_foliation' to pttype; &r dip.aml
'Meta Vert Fol'  new; moveitem 'metamorphic_vertical_foliation' to pttype; &r dip.aml
'Meta Horz Fol'  new; moveitem 'metamorphic_horizontal_foliation' to pttype; &r dip.aml
'Mylonitic Fol'   new; moveitem 'mylonitic_foliation' to pttype; &r dip.aml

PLUNGE
'Lineation'       new; moveitem 'lineation' to pttype; &r plunge.aml
'Mineral Lin'     new; moveitem 'mineral_lineation' to pttype; &r plunge.aml
'Mylonitic Lin'   new; moveitem 'mylonitic_lineation' to pttype; &r plunge.aml
'Slicks Fill'     new; moveitem 'slickenlines_fill' to pttype; &r plunge.aml
'Slicks Open'     new; moveitem 'slickenlines_open' to pttype; &r plunge.aml
'Flt Dip Fill'   new; moveitem 'fault_dip_fill' to pttype; &r plunge.aml
'Flt Dip Open'   new; moveitem 'fault_dip_open' to pttype; &r plunge.aml
'Shear Sense'     new; moveitem 'shear_sense' to pttype; &r plunge.aml

OTHER
'Minor fold'     new; moveitem 'minor_fold' to pttype; &r dip.aml
'Sfold'           new; moveitem 'sfold' to pttype; &r dip.aml
'Zfold'           new; moveitem 'zfold' to pttype; &r dip.aml
'Dike box'        new; moveitem 'dike_box' to pttype; &r dip.aml

COORDINATE
'Mouse'           coordinate mouse
'Digitizer'       coordinate digitizer
'Enable Summa'    &sv user = [username]; ~
                  digitizer sgm93 /tmp/.dig_signal_%user%:9600:7:even
'Enable Dig'      digitizer altek /dev/tty00:9600:8bit:none
'Dig Cover'       coordinate digitizer %cover%

EF
'efAnno'          ef anno
'efPoint'         ef point
EF

DRAG
ADD ANNO
      textset font; annosymbol 2; ef anno; ~
      &sv dipheight_in = .0833; ~
      &sv dipheight_m = [round [calc %dipheight_in% / 39.37 * %.mapscale%]]; ~
      annofeature point dip; annosize %dipheight_m%; ~
      add new; annoselectfeature *; ~
      annotext feature; annoplacement feature

DRAW
'Draw'            draw
'Draw Noclear'   draw noclear

PAGEEXT
'Zoom'            pageextent *; draw
'Fullview'        pageextent page; draw

SELECT

```

```
'Sel One'      select
'Sel Many'     select many
'Sel Box Pass' sel box pass
'Sel Pttpe'    &sv pttpe = [response 'Enter pttpe']; ~
                select pttpe = %pttpe%
DELETE
  Delete      delete
'Del One'     sel; delete
'Del Many'    sel many; delete
'Del Box'     sel box pass; delete
LIST
  List        list
>List$        list $all
'List One'   sel; list
'List Many'  sel many; list
'List $One'   sel; list $all
'List $Many'  sel many; list $all
EDITUNIT      &return; &r unitanno.aml
OOPS          oops
SAVE          save; &type ALL SAVED!
QUIT
  Quit       &return; quit
'End and Build' &return; quit; build %cover% point
```

```

/* PLUNGE.AML by Haydee Hampton 6/12/97
/* Arcedit AML for digitizing trend and plunge symbol.
/* Modified by D.Block 6/98 to add annotation.

/* User digitizes plunge arrow and enters plunge value.
/* AML adds a point with the correct $ANGLE for marker rotation and
/* STRIKE item. Plunge angle is stored in DIP item.

/* Trend (stored in STRIKE item) angle is calculated based on the right hand-rule.

/* Hit "9" during ADD ONE to end AML session.

/* BEFORE USING THIS AML (H.Hampton):
/* 1. Add POINTTYPE, DIP and STRIKE items to the .PAT file
/* 2. Build with the line option without adding any items.
/* (Reason: This AML uses the .aat file to calculate length of arcs)
/* 3. Set the POINTTYPE to lineation
/* with the NEW command after adding strike and dip symbol with DIP.AML.

/* BEFORE USING THIS AML (D.Block):
/* 1. Run NEWLUT.AML to add lookup tables and relate file to workspace (if not there).
/* 2. Run ADDITEMDIP.AML to create coverage and add items to .pat file.
/* 3. Run ATTITUDE.AML to invoke menu which incoporates this AML.

&type Beginning PLUNGE.AML

&do &while %.plungeflag% <> 0 /* loop for adding more than one marker

/* Use "coordinate digitizer" command if using digitizing tablet.
/* Use "coordinate mouse" command below if using mouse. Do not use both.

coordinate digitizer
/*coordinate mouse
textset font
de arc arrows point anno
ef arc

/* Digitize plunge arrow from where it begins on the strike and dip to its arrowhead

add one

/* Bail out of AML if user hits "9" during the arc digitizing session
/* initiated by the ADD ONE command above

&if [show number selected] = 0 &then &do
  coordinate mouse /* return coordinate entry to mouse
  ef point
  &type End of PLUNGE.AML
  &return
  &end
&else

/* Extract coordinate values of from and to node of selected arc

&setvar .x1 [extract 1 [show arc [show select 1] nodes]]
&setvar .y1 [extract 2 [show arc [show select 1] nodes]]
&setvar .x2 [extract 3 [show arc [show select 1] nodes]]
&setvar .y2 [extract 4 [show arc [show select 1] nodes]]

/* Calculate length of arc in order to place marker
/* at from node of digitized arc

&sv .len [extract 1 [show arc [show select 1] item length]]; &type .len = %.len%

/* Calculate the marker angle and
/* add a marker point at the from node of digitized line

ef point
&setvar angle = [radang [invangle %.x1% %.y1% %.x2% %.y2%]]
coordinate keyboard polar; &type coordinate keyboard polar
add one

```

```

1 %angle% -%.len%
/* Assign a value to the $ANGLE pseudoitem

&if %angle% lt 90 &then
    calc $angle = [radang [invangle %.x1% %.y1% %.x2% %.y2%]] + 270
&else
    calc $angle = [radang [invangle %.x1% %.y1% %.x2% %.y2%]] - 90

/* Assign a value to the DIP item.
/* Dip may be entered from the puck or the keyboard.
/*
/* &type Enter dip angle using puck keypad, then enter "A"
/* calc dip = [dignum]

calc dip = [response 'Enter dip angle using keyboard']

/* Assign a value to the strike item

&sv strike = [mod [round [calc 450 - %angle%]] 360]
calc strike = %strike%

/* Display dip value as annotation

&sv dipheight_in = .0833 /* 6 point
&sv dipoffset_in = .11
&sv dipheight_m = [round [calc %dipheight_in% / 39.37 * %.mapscale%]]
&sv dipoffset_m = [round [calc %dipoffset_in% / 39.37 * %.mapscale%]]

&sv pi = 3.1415927
&sv theta = [round [calc 90 - %strike%]] /* complement of strike
&sv rads = [round [calc %theta% / 360 * 2 * %pi%]] /* convert from degrees to radians
&sv 3x = [round [calc ( %.x1% + %.x2% ) / 2]] /* 3x,3y = midpoint of arc or symbol
&sv 3y = [round [calc ( %.y1% + %.y2% ) / 2]]
&sv ax = [round [calc %3x% + %dipoffset_m% * [sin %rads%]]]
&sv ay = [round [calc %3y% - %dipoffset_m% * [cos %rads%]]]

textset font
annosymbol 2 /* Univers Medium Italic

ef anno
annofeature point dip
annosize %dipheight_m%
/*coordinate cursor
/*add new; &type Select symbol to annotate, then enter 9 on keyboard to continue
/*coordinate digitizer
coordinate mouse
add new; &type Select symbol to annotate using #1 key, then enter 9 on keypad to continue
annoselectfeature
annotext feature
annoplace feature
&pushpoint 1 %ax% %ay%
reposition

/* Display .PAT items on screen

&type \$ANGLE = [radang [invangle %.x1% %.y1% %.x2% %.y2%]]
&type STRIKE = [mod [round [calc 450 - %angle%]] 360]
/*&type DIP = [show label [show select 1] item dip]

/* Delete digitized arc

ef arc
coordinate keyboard xy
sel one x: %.x1% %.y1%
delete

coordinate mouse
ef point

&end

```



```

/* Haydee Hampton 5/19/97
/* Arcedit AML for digitizing bedding and foliation symbols.
/* Modified by D.Block 6/98 to add annotation.

/* User digitizes left and right ends of strike and enters dip value.
/* AML adds a point with the correct $ANGLE for marker rotation and
/* STRIKE item.

/* Strike angle is calculated based on the right hand-rule.

/* Hit "9" at arc digitizing prompt (initiated by ADD ONE) to end AML session.

/* BEFORE USING THIS AML (H.Hampton):
/* 1. Add POINTTYPE, DIP and STRIKE items to the .PAT (point attribute
/* table).
/* 2. Build with the line option without adding any items.
/* (Reason: This AML uses the .aat file to calculate length of arcs)
/* 3. Set the POINTTYPE (e.g., bedding, magmatic foliation, etc.)
/* with the NEW command each time it changes. Set markers using
/* MARKERSET command. Set up relate environment between point marker LUT
/* and .PAT

/* BEFORE USING THIS AML (D.Block):
/* 1. Run NEWLUT.AML to add lookup tables and relate file to workspace (if not there).
/* 2. Run ADDITEMDIP.AML to create coverage and add items to .pat file.
/* 3. Run ATTITUDE.AML to invoke menu which incoporates this AML.

&type Beginning DIP.AML

&do &while %.termflag% <> 0 /* Continuous loop for adding more than one marker.
                                /* Loop ends when user hits "9" to terminate
                                /* ADD ONE arc digitizing session.

/* Use "coordinate digitizer" command if using digitizing tablet.
/* Use "coordinate mouse" command below if using mouse. Do not use both.

/*coordinate digitizer
coordinate mouse
textset font
de arc arrows point anno
ef arc

/* Digitize arc from "first" to "second" end of bedding symbol
/* (the first side of the bedding symbol is on left when dip is pointing up)

add one

/* Bail out of AML if user hits "9" during the arc digitizing session
/* initiated by the ADD ONE command above

&if [show number selected] = 0 &then &do
  coordinate mouse /* return coordinate entry to mouse
  ef point
  &type End of DIP.AML
  &return
  &end
&else

/* Extract coordinate values of from and to node of selected arc

&setvar .x1 [extract 1 [show arc [show select 1] nodes]]
&setvar .y1 [extract 2 [show arc [show select 1] nodes]]
&setvar .x2 [extract 3 [show arc [show select 1] nodes]]
&setvar .y2 [extract 4 [show arc [show select 1] nodes]]

/* Calculate length of half the arc in order to place marker
/* at midpoint of arc

&sv .len [extract 1 [show arc [show select 1] item length]]; &type .len = %.len%
&sv .halflen = %.len% / 2; &type .halflen = %.halflen%

```

```

/* Calculate the $ANGLE pseudoitem for the marker and
/* add marker point at midpoint of digitized line

ef point
&setvar angle = [radang [invangle %.x1% %.y1% %.x2% %.y2%]]
coordinate keyboard polar; &type coordinate keyboard polar
add one
1 %angle% -%.halflen%

/* Assign a value to the $ANGLE pseudoitem

calc $angle = [radang [invangle %.x1% %.y1% %.x2% %.y2%]]

/* Assign a value to the DIP item.
/* Dip may be entered from the puck or the keyboard.
/*
/* &type Enter dip angle using puck keypad, then enter "A"
/* calc dip = [dignum]

calc dip = [response 'Enter dip angle using keyboard']

/* Assign a value to the STRIKE item for the next marker point

&sv strike = [mod [round [calc 630 - %angle%]] 360]
calc strike = %strike%

/* Display dip value as annotation

&sv dipheight_in = .0833 /* 6 point
&sv dipoffset_in = .10 /* 7.2 point
&sv dipheight_m = [round [calc %dipheight_in% / 39.37 * %.mapscale%]]
&sv dipoffset_m = [round [calc %dipoffset_in% / 39.37 * %.mapscale%]]

&sv pi = 3.1415927
&sv theta = [round [calc 90 - %strike%]] /* complement of strike
&sv rads = [round [calc %theta% / 360 * 2 * %pi%]] /* convert from degrees to radians
&sv 3x = [round [calc ( %.x1% + %.x2% ) / 2]] /* 3x,3y = midpoint of arc or symbol
&sv 3y = [round [calc ( %.y1% + %.y2% ) / 2]]
&sv ax = [round [calc %3x% + [sin %rads%] * %dipoffset_m%]] /* 60
&sv ay = [round [calc %3y% - [cos %rads%] * %dipoffset_m%]] /* 60

textset font
annosymbol 2 /* Univers Medium Italic

ef anno
annofeature point dip
annosize %dipheight_m%
coordinate cursor
add new; &type Select symbol to annotate, then enter 9 on keyboard to continue
annoselectfeature
annotext feature
annoplace feature
&pushpoint 1 %ax% %ay%
reposition

/* Display .PAT items on screen

&type ANGLE = [radang [invangle %.x1% %.y1% %.x2% %.y2%]]
&type STRIKE = [mod [round [calc 630 - %angle%]] 360]
&type DIP = [show label [show select 1] item dip]

/* Delete digitized arc

ef arc
coordinate keyboard xy
sel one x: %.x1% %.y1%
delete

coordinate mouse
ef point

```

&end

ATTACHMENT 13

```
/* ADDITEMANNO.AML
/*
/* Creates anno coverage from selected polygon coverage

&sv bndcov = [getcover * -poly 'Select poly cover with unit names in .pat']
&sv cover = [response 'Enter name of new anno cover']
&sv mapscale = [response 'Enter the mapscale']
&sv asize_in = .1111 /* 8 point
&sv asize_m = [round [calc %asize_in% / 39.37 * %mapscale%]]

copy %bndcov% %cover%

AP
mape %cover%
annocoverage %cover% unit
textset font
textsymb 1 /* Univers Medium
labeltext %cover% ptype
quit

AE
ec %cover%
ef anno.unit
sel all
calc $fit = 'off'
calc $size = %asize_m%
ef poly
sel all
delete
save
quit

build %cover% line

&return
```

ATTACHMENT 14

```
/* ANNOUNITEDIT.AML
/* AML for editing unit name annotation; run from ARC prompt.
/* Display set for 20 inch monitor.

display 9999 size 1280 650 pos 100 75
ae
&sv cover = [GETCOVER * -anno 'Select an anno coverage']

mape %cover%
mapunits meters
mapscale [RESPONSE 'Enter the mapscale']
ec %cover%
snapping off

lineset geo.lin
textset font
markerset geo.mrk

relate restore geo.rel
symbolitem %cover% arc arcrel//symbol
de anno.unit arc
ef anno.unit

&sv bcov1 = [GETCOVER * -poly 'Select a poly backcoverage' -none]
&if NOT [NULL %bcov1%] &then
  &do
    bc %bcov1%
    backsymbolitem %bcov1% arc arcrel//symbol
    be %bcov1% arc
    ap [GETFILE *.mrk.aml 'Select to display arcmarkers' -none]
  &end
&sv bcov2 = [GETCOVER * -point 'Select a dip backcoverage' -none]
&if NOT [NULL %bcov2%] &then
  &do
    bc %bcov2%
    backsymbolitem %bcov2% point dipptrel//symbol
    be %bcov2% point anno
  &end
&sv bcov3 = [GETCOVER * -arc 'Select a fold backcoverage' -none]
&if NOT [NULL %bcov3%] &then
  &do
    bc %bcov3%
    backsymbolitem %bcov3% arc arcrel//symbol
    be %bcov3% arc
  &end

setdrawsymbol 0 magenta
draw

&terminal 9999
&fullscreen &popup
&menu /arcwork/menus/announit.menu &stripe announit.menu

&return
```

```

1 announit.menu
COMMAND          &sv command [RESPONSE 'Type Command']; ~
                  &lv command; ~
                  [UNQ %command%]
SEL              sel
'SEL MENU'
'Sel Text'       &sv txt [RESPONSE 'Enter text']; sel $text = %txt%; drawselect
'Sel Many'       sel many
'Sel Box'        sel box
'Sel Box Pass'  sel box passthru
'Sel All'        sel all
'Asel Many'     asel many
'Unsel'          unsel
'Unsel All'     unsel all
'Nsel'           nsel
DELETE           sel; delete
'DEL MENU'
'Del Many'       sel many; delete
'Del Box'        sel box; delete
'Del Poly'       sel poly; delete
'Del Box Pass'  sel box passthru; delete
DRAG             sel; drag
'ADD ANNO'
COPY             add new; annotype point1; annopos cc; ~
                  &sv anno [RESPONSE 'Type annotext']; annotext [UNQ %anno%]; ~
                  &sv asize_in = .1111; ~
                  &sv asize_m = [round [calc %asize_in% / 39.37 * %.mapscale%]]; ~
                  calc $size = %asize_m%;
LIST             sel; copy
'List One'       sel; list
'List Many'      sel; list many
'List $One'      sel; list $all
'List $Many'     sel many; list $all
ZOOM             pageextent *; draw
FULLVIEW         pageextent page; draw
AP
'AP Color'      ap [GETFILE *.ap.aml 'Select an ap file']; apmode edit; draw
'AP None'        ap none
'ADD ARC'        ef arc; add; calc ltype = 'leader'; draw
EF
    Anno          ef anno.unit
    Arc           ef arc
EDITDIP          &return; ef anno; &menu attitude.menu &stripe attitude.menu
OOPS             oops
DRAW
    Draw          draw
    'Draw Noclear' draw noclear
SAVE             save; &type ALL SAVED!
QUIT             &type SO LONG!; &return; quit

```

